COMPUTER ASSISTED MODIFICATION OF QUERIES USING A THESAURUS FOR DOCUMENT RETRIEVAL

by

Guy Agoston

A Thesis In

The Department of

Computer Science

Presented in Partial Fulfillment of the Requirements

for a Degree of Master of Computer Science at

Concordia University

Montreal, Quebec, Canada

May 1981

ABSTRACT

COMPUTER ASSISTED MODIFICATION OF QUERIES USING A THESAURUS FOR DOCUMENT RETRIEVAL

Guy Agoston

One of the several reasons for imperfection (low recall or low precision) in document retrieval lies with the user specification of search terms. This problem is more pronounced when the vocabularies of indexers and users are not controlled. A possible way of improvement is to enhance or augment the set of search terms initially specified by the user.

A sample of 480 documents were examined from the Communication of the ACM (CACM), spanning 5 years. High frequency words were extracted from the title, abstract and key-word sections of each document. These high frequency words were then associated with the published categories of the classification system for Computing Reviews (CR categories). Each of these CR categories constitute a cluster of terms, relevant to that category. It was presumed that certain terms will appear in more than one cluster, and that this cluster overlap could be utilized in an interactive system, where the user could define the meaning of a term relevant to his query.

After defining a single or multiple meaning (category), the contents of the cluster would be printed for the user's query enhancement. The user would then choose from these related terms and include them in his modified query, which should yield higher recall and higher precision.

The collected clusters from the CACM texts were then compared term by term to the corresponding categories of the known National Computing Center (NCC) thesaurus. The comparison between the two sets were to indicate:

1 how representative the collected clusters from the CACM were,

2 whether these clusters are more/less representative than the base they were compared to,

3 whether these clusters would represent an enhancement to the original query, if related terms were selectively inserted into the query.

The significance of different configurations of term frequencies between the title, the abstract and key-words were examined also.

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to my supervisor T. Radhakrishnon for his advice, criticism and guidance throughout the duration of this research. I want to thank him for always finding the time to help me in spite of his very tight schedule.

TABLE OF CONTENTS

FIGURES

TABLES

CHAPTER I

INTRODUCTION

1.1 INFORMATION SYSTEMS

A natural outcome of the information explosion of the sixties was the development of automatic information systems. The basic purpose of an information system is to place the user in more efficient and direct contact with the data bases of concern to him and thus to enable him to retrieve information. The Annual Review of Information Science and Technology (INF,74) defined information retrieval as the following. "A system capable of retrieving information relating to documents, usually a representation or surrogate of a document (e. g. title or abstract) in response to a question, and the detection of relevant or near-relevant documents (or references to documents) that either answer the question in themselves or that can be further scanned to determine the answer."

The words "document retrieval" and "information retrieval" will be used interchangeably in this thesis, as is generally the case in the literature.

Various tools have been developed to improve the efficiency of information retrieval. The techniques of coordinate indexing, the employment of classification schemes, the development of thesauri, the batching of computer profiles, and on-line query languages are all aids of this kind. Different techniques for the enhancement of user query are also aids to improve the efficiency of information retrieval. The work described in this thesis focuses on various methods of query enhancement using on-line thesaurus.

The on-line manipulation of the thesaurus requires careful consideration. The thesaurus must somehow be linked to the data base of mixed bibliographic format to facilitate both the indexing of the new data base entries and the searching of existing data base entries for information. It must also be linked to classification codes to allow for searching of the data base with class numbers as well as keywords. Questions arise about what relationships should be permitted in the thesaurus and about how many entries should be allowed under a specific relationship for a given entry. The central concern is that the user, through the medium of a suitable query language, should be able to create, modify and display a thesaurus or parts of it and this must be done at a reasonable price in computer memory and response time.

The main themes of the thesis may be grouped in a general way and they will be discussed in the following order.

1. The thesaurus concept and literature concerned with it;
2. the description of the query enhancement methodology, the sample used and the results obtained;
3. recommendation and summary.

1.2 THE THESAURUS CONCEPT

Thesaurus is a Greek word and it means "storehouse" or "treasury". Reader's Digest described their recently published Thesaurus of Family Word Finder (REA,77) as a synonym dictionary, a dictionary backwards, which is being used when one knows the meaning of a word, but is looking for another, more precise word which could be used in a given context. B. C. Vickery (VIC,66) believes that thesaurus can have two meanings:

1. any linear list displaying relations between words, and
2. a tool aiding us to pass from text words in a natural language to keywords or codes in a standardized language.

His definition seems to satisfy both the computer and non-computer worlds. His first definition will be recognized as valid by anyone who has seen ROGET'S THESAURUS. The Reader's Digest Family Word Finder Thesaurus would be another example for the above. The idea of a word list with defined relationships

between terms is implied in this definition.

The second meaning is not readily understood by the layman. The information specialist, however, who uses a thesaurus for indexing and searching understands the definition and perceives it to be correct. The second part of the definition indicates the reason for research into the feasibility of totally automated or computer assisted thesaurus construction; this statement in no way undermines the associated or equal importance of the first meaning.

According to the UNESCO 1970 GUIDELINES (UNE,70), a thesaurus is defined as "a controlled and dynamic vocabulary of semantically and generically related terms which comprehensively covers a specific domain of knowledge. This vocabulary is a systematic and/or alphabetic collection of descriptors, non-descriptors (auxiliary terms) as well as indicators of their relationship".

J. C. Costello Jr. (COS,66) has defined the thesaurus in a manner that reinforces the above definitions, moreover, it stresses the aspects that are important in this thesis. He says: "By definition, an information retrieval thesaurus is a

> display of unit concept terms of an index vocabulary in which terms are alphabetically ordered and in which the relationships of each term to the other terms in the index vocabulary are systematically presented".

Instead of thinking of a thesaurus in its definitional sense, that is, a dictionary of synonyms and antonyms, it may be useful to think of it as a compilation which contains the terms of a given information retrieval system's vocabulary, arranged in some meaningful form, and which provides information relating to each term that will enable the user of the information file to predict the relevance of responses to questions when this particular vocabulary control is used.

The user of an information retrieval system, either manual or automated, wants to be reasonably certain that the queries he prepares will retrieve information which is relevant to his interest. A search conducted with user chosen keywords on a document collection or data base may, or may not, retrieve information related to the user's interests. Obviously a method by which the user could be assured, to some degree, that the retrieved information is relevant to his interests would be an invaluable aid in preparing queries for a retrieval system. One such aid is through vocabulary control using thesaurus.

It is evident that a controlled vocabulary, such as a thesaurus, will be used with many types of data bases particularly when one or more of the following conditions exist:

1. the data base covers a wide range of subject matter,
2. the potential users have different backgrounds and information requirements,
3. the data base shows a lack of continuity in term

usage, term meanings, and physical appearance of terms (noun forms, plurals, punctuation)

4. the user is unable to determine all possible avenues of searching for the information. This often occurs if the indexer and searcher are not the same person.

Discussing an information storage and retrieval system, Costello has also given reasons for vocabulary control. These reasons are closely related to the above conditions when they apply to an information retrieval system. His reasons for vocabulary controls are:

1. to improve the quality of description of document content at the time of input
2. to improve the quality of description of desired contents at the time of output
3. to improve the relevance and recall ratio characteristics of the system.

Thus, in information storage and retrieval systems that operate with keyworded information, a thesaurus should be employed in the indexing of documents.[4] The same thesaurus should also be used in preparing queries to the system. The queries then will consist of, for the most part, the same keywords that are used in indexing the documents or information pertinent to the user's interests. We can see therefore the twofold functions of a thesaurus: one is as a tool for regulating the output, the other is as an autority list for the

input.

K. Sparck Jones (SPA,70) states that: " ..it is generally true that the stronger the match between a request and a document, the more chance the document has of being relevant to the request...".

Thus the use of a thesaurus in both indexing and query formulation provides a congruence between indexing and searching languages. This agreement, in most cases, improves retrieval effectiveness.

The mode of thesaurus construction also has a decisive effect on retrieval effectiveness. When a thesaurus is manually created, it usually exists in printed form, which could be quite voluminous. To locate a wanted item of information can be hard, tedious and discouraging for the users. The updating of such a thesaurus usually requires major reindexing and reprinting, which can be both time consuming and expensive. To justify the expenses, the changes are accumulated over some time period for an update. By its character therefore, a manually constructed thesaurus tends to be static in nature, and its timeliness could be in question.

With new techniques and modern computing equipment, a thesaurus can be stored as a computer file, thus transforming it into a potentially dynamic entity. Procedures are available for organizing and analyzing the information in storage, and

real-time software and hardware can be used to ensure that the stored information is retrieved in response to requests from a user population, in a convenient form and at little cost in time and effort.

In today's time sharing environment, the basic operations performed are searches conducted on data bases. Often the type of search conducted is a weighted term search with "questions" to the system consisting of user specified terms, which hopefully are the same as those used to index the documents or information that the user wishes to obtain. In this type of system, the thesaurus may often be absolutely necessary to insure optimum matching of these search questions. If such information systems are to operate in on-line mode, there is no reason why a computer program could not be constructed to allow a user to develop and manipulate a thesaurus on-line. The main concern of this thesis is the on-line manipulation of thesaurus for query enhancement.

1.3 THE THESAURUS FORMAT AND RELATIONSHIPS

Information retrieval thesauri are used in both the indexing and searching phases of the retrieval system. In these, thesauri terms are linked together by indicated relationships. So called "main terms" are listed in alphabetic order and under each main term usually a series of "other terms" are listed. Generally these other terms are marked as "related

terms", "narrower terms", "broader terms".

An explanation of some of the relationships in common use today were taken from the National Computing Center Thesaurus (NCC,73) and from the Thesaurus of Information Science Terminology (THE,68).

1. SCOPE note - This is a short explanation intended to clear any ambiguity in the meaning. It indicates how the term is used and gives a brief description.

2. USE - This relationship usually indicates that the main term is not acceptable as an indexing or searching keyword and that the term following the "use" should be used in its place. Examples of categories of terms that are not "used" are:

T_2 USE T_1

- true synonyms
  e.g. Aerials USE Antennas
- very specific terms where a broader term is used
  e.g. Drums (memory) USE Storage Media
- abbreviations
  e.g. CAI USE Computer Aided Instruction

3. UF (Used for) - This is the inverse of the USE relationship. The terms following the UF are the terms which have been referenced to the "main term" under which the UF reference appears. An example will

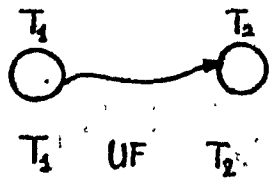
illustrate the USE and UF references.

DISC

USE : STORAGE MEDIA

STORAGE MEDIA

UF : DISC



4. BT (Broader term) - The terms following this reference represent a broader class of concepts to which the main term belongs. The references give a more general term associated with the main term and thus allow for more general indexing or searching.

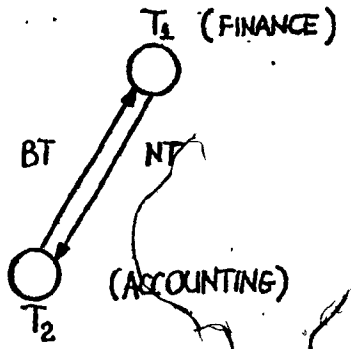
5. NT (Narrower term) - The terms listed are generally narrower than the main term. These references give a more specific term associated with the main term and thus facilitate more specific indexing or searching. The BT and NT references are illustrated below.

ACCOUNTING

BT : FINANCE

FINANCE

NT : ACCOUNTING



6. RT (Related term) - The terms following this reference are related to the main term conceptually but not hierarchically. These references are means to broaden

the scope of indexing or searching. The RT reference is illustrated below.

AUDIO RESPONSE SYSTEM

RT : SPEECH RECOGNITION

SPEECH RECOGNITION

RT : AUDIO RESPONSE SYSTEM

7. SYNONYMS - The terms following this reference are synonyms with the main term under which they are listed. In many thesauri, "synonyms" are not indicated, but synonyms or near synonym terms are marked with the USE reference, which gives the user the preferred term to use in indexing or searching.

The above are the most commonly used relationships in modern thesauri.

1.4 OUTLINE OF THE THESIS

A brief survey of the literature concerned with thesaurus construction is described in chapter II. The manual, semi-automatic and automatic construction methods are detailed. Chapter III discusses the major elements of information retrieval such as file structures and access strategies, measurement of relevance and the different type of information systems. Chapter IV contains a description of document vector space and the proposed approach for query enhancement. It describes the study conducted and presents the experimental findings. Chapter V, the last chapter, deals with the conclusions derived from these findings.

CHAPTER II

LITERARY REVIEW - THESAURUS CREATION

This chapter will cover the literature that is concerned with thesaurus construction. The manual, automatic and semi-automatic construction methods will be described with their inherent problems. The literature describing various thesaurus organizational techniques will also be discussed.

2.1 THEORETICAL CONSIDERATIONS

Thesaurus construction has basically two aspects:

1. selecting the keywords/phrases of the particular subject for which the thesaurus is constructed, and
2. establishing an inter-relationship among those terms which deal with the problem of synonymity and hierarchical relations among terms.

The importance of both these aspects for information retrieval purpose is evident. There are two approaches to thesaurus construction:

1. "a priori" classification or faceted method, and

2. "a posteriori" or statistical method based purely on frequency of terms.

In traditional classification theory, certain terms characterize important branches of a subject, some terms specify the important topics within the different branches, and there can be still other terms for the different sub-topics and so on. The selection of these terms and hierarchical relations among them could be considered as a first step towards thesaurus construction. If such a thesaurus is to be of use for information seekers, then it is assumed that the important topics and sub-topics of the subject are uniquely determined by terms which are well known to users of the system. Due to modern interdisciplinary development of knowledge, it has almost become impossible to characterize an important topic just by a few terms. The topic may have many "facets". It may be important from different points of view. For instance, a text on quantum mechanics may be important not only for nuclear physics, but also for mathematics, logic and philosophy. These many "facets" of a topic were discussed by Vickery (VIC,68).

C. H. Davis (DAV,68) discussed the linking of vocabularies and classification schemes derived "a priori" (in advance) from these many facets. Davis contends that thesaurus entries are not as important as the thesaurus entries included with their associated relationships. In the classification schedules, the classification numbers would be followed by the

corresponding terms. In application of this to a retrieval system, the document would be indexed by classification number instead of terms. Matching for retrieval would then be done on classification numbers.

A faceted classification scheme and a thesaurus covering engineering, scientific, technical, and management subjects have been integrated to form THESAUROFACET. As described by J. Aitchison (AIT,69), the terms appear in both the classification schedules and in the thesaurus and they are linked by the notation or class number. The thesaurus serves as an index to the classification schedules. The Thesaurofacet is a multipurpose tool, it can be applied to coordinate indexing, used in computerised retrieval systems, and classifications. It is a non-computerised approach to the concept of linking a thesaurus and a classification scheme together.

In semi and fully automatic thesaurus construction, usually "a posteriori" selection processes are employed.

One such process, selection of key words, uses the frequency of terms found in a given collection of documents and then chooses as significant those terms for which the frequencies are neither too high nor too low (SAL,75), (SPA,71). These words are then arranged in a sequence according to the magnitude of their relative frequencies. At this point, it is expected that the terms important for information retrieval would cluster around the middle of the frequency range. Those

frequencies too low or too high can be considered to be insignificant for information retrieval. The terms of very high frequency are considered as common words and those of very low frequency are considered to have little relevance for the subject. The decision as to the limits, however, remains arbitrary. Statistical methods are never meant to give exact results. If, most of the time, the vocabulary of the user approximately coincides with the "treasure" of keywords determined by eliminating words "too low" and "too high" in frequency, then the method of eliminating high and low frequency words can be accepted for thesaurus construction.

Another method uses the inter-relationship of terms by means of a term document matrix based on the frequency theory. G. Salton (SAL,68) in his SMART system has done much in this area. The frequency of terms in each document belonging to a collection of document on the subject for which a thesaurus is to be made is represented in a matrix form. See figure 2.1.

The ij-th element represents the number of times term j appears in document i. In this example we find that t1, t4 are highly frequent in d1 and d2, whereas in d3 and d4 the frequency of both these terms is low. Similarly for t2, t3 we find that they are highly frequent in d3, d4 and rare in d1, d2. Hence, one may conclude that t1 is related to t4 and t2 is related to t3. In this way, classes of inter-related terms are formed so that terms belonging to the same class are characterised by the

fact that they are almost equally frequent in a set of documents and almost equally rare in another set of documents. As in the selection of key words, the relationship among the terms is established purely statistically.

The frequency approach has also been used to set up hierarchical relationships among terms. As Doyle (DOY,65) explained, for a class of interrelated terms, those terms which have substantially higher frequencies are considered as categories and those with low frequencies are considered as subcategories. Hierarchical relationships based on frequency of terms might have been adequate for the information needs of a particular subject, but there does not seem to be enough ground to make it a general principle for thesaurus construction for all subjects. As admitted by Salton (SAL,68), setting up hierarchical relationships of terms requires certain amount of human judgement. The question is how human judgement is to be used for establishing a classification as well as an inter-relationship of terms.

"Question - Answering Systems" (SAL,68) have been used to apply human judgement to the problem of classification of terms and determination of categories and subcategories. The categories and subcategories are given "a priori", such as abstract, concrete, etc., and then questions are asked whether the terms have the properties expressed by the categories. In this way, one gets a term property matrix (see figure 2.2).

	t_1	t_2	t_3	t_4
d_1	4	1	0	5
d_2	3	0	1	4
d_3	0	4	6	1
d_4	1	3	4	0

TERM DOCUMENT MATRIX

FIGURE 2.1

	Abstraction	Physical Object	Hardware	Software
Computer	0	1	1	0
System				
Program				
Machine				
Equation				
Logic				
Data				

TERM PROPERTY MATRIX

FIGURE 2.2

The properties are chosen in such a way that they are apparently mutually exclusive. For the term "computer" there is no ambiguity as to which of the properties would apply. It would be: 0110, i.e., computer is not an abstraction, it is a physical object, it is hardware and not software. For the terms "system" and "program", it is not quite clear which of the properties apply. For instance program can be considered as an abstraction, if it is thought of as a mathematical algorithm, and on the other hand as a physical object if considered as a "written thing". Similarly, system may be both software and hardware. Sometimes a term may be such that a property is neither applicable nor non applicable to it. It may be quite meaningless to ask whether the term "data" is software or hardware.

If one could choose such properties which are unambiguously mutually exclusive, then one would also be able to set up a hierarchical classification of terms from the term property matrix. This is dependent on the subject matter, for the terms in natural sciences are determined in terms of physically controlled experiments and the artificial language of mathematics; there is much less ambiguity about them than for the terms of social sciences. Hence the construction of a thesaurus for engineering or atomic physics will present much less problem than that of a thesaurus for sociology or political science.

2.2 MANUAL CONSTRUCTION

The manual thesaurus construction is still the most common method used today, although, when automatically generated keywords are compared with terms manually assigned by subject experts, one normally finds agreement for 60% to 80% of the assigned terms (SAL,73). It is necessary to develop a set of carefully prepared instructions specifying the required steps and setting forth in detail the meanings and implications of choosing one or another of the permissible alternatives. These carefully prepared instructions are based on human judgement. A good example of this is the "Manual on the Construction of an Indexing Language" (CRO,71) which describes the indexing techniques and the construction of thesaurus.

Some thesauri using manual construction methods will be now described.

In its introduction on page vii (WAT,66), The Water Resources Thesaurus says that it was: "prepared by qualified scientists who carefully processed lists of candidate terms to determine their general utility for describing water resources research and development efforts and to identify the semantic relationships among them".

The terms consisted of those used in the preparation of the

Catalog of Water Resources Research and those suggested by scientists, engineers and other specialists.

The method used in compiling the Thesaurus of Pulp and Paper Terms (PUL,65) is an illustration of another way in which a thesaurus can be prepared. Using internal indexes at the Pulp and Paper Research Institute, a first draft was compiled. Then more terms were included by references to bibliographic indexes. This draft thesaurus was then used to index abstracts from the Abstract Bulletin of the Institute of Paper Chemistry. This procedure generated additional keywords and also pointed out deficiencies in the original draft. A committee then looked at each term individually, considering its utility, ambiguity in meaning, and cross relationships with other terms. The method used in compiling the Thesaurus of Pulp and Paper Terms is a good one. By applying the draft thesaurus to the type of situation in which it would undoubtedly be used, the inadequacies of the draft were brought to light and changes were made before final printing took place.

Other thesauri were produced by a series of steps which closely adhered to the steps followed in the compilation of the Thesaurus of Pulp and Paper Terms. The main difference is that these thesauri were initially published and used in indexing and searching. From the experience drawn from use of these thesauri, changes were made and updated issues were published. Two thesauri which were prepared in this manner are the

Thesaurus of Textile Terms and the Thesaurus of ERIC Descriptors (First Edition) of the Education Resources Information Center (ERI,68).

The office of Naval Research (ONR) was assigned the task to create a Technical Thesaurus for the United States Department of Defence. Besides creation of the actual thesaurus, one of the project requirements was to prepare a manual which indicated the method used in building the thesaurus. This manual turned out to be an excellent guide, which could be used in building almost any technical thesaurus. The manual covers almost everything involved in thesaurus construction, including fundamental term rules, cross reference rules and alphabetization rules. About 35 people participated in deliberations which resulted in the development of the manual. The thesaurus produced was called TEST (Thesaurus of Engineering and Scientific Terms) (TES,69).

The majority of thesauri created by manual methods exist in printed form either as a book or as unbound pages. Periodically the thesaurus will require updating. This almost certainly means that major reindexing will be required, hence reprinting will be necessary, which can be both time consuming and expensive. Because of this, the timeliness of these thesauri should be questioned. The second criticism of this method is more fundamental. The opinions and actions of a small group tend to follow those of the most dominant members of the group. Although a committee set up to choose terms for a thesaurus

would be very carefully chosen, the previously mentioned situation may well occur. Nevertheless, the advantages of using this approach should be obvious. If a representative cross section of people consider the utility of candidate terms, in theory the thesaurus should have no special bias. This method of thesaurus construction has been the one most often used over the years and undoubtedly in the future it will continue to be widely used.

2.3 SEMI - AUTOMATIC CONSTRUCTION

Semi-automatic methods are generally based on various automatic aids and subject experts for the basic task of defining the meaning of each term being introduced into the thesaurus. The basis for thesaurus entries is a word frequency list usually generated by automatic means. By answering certain questions, the terms can be represented by a property matrix (see figure 2.2). The rows of this matrix can be manipulated; identical rows can be combined; by eliminating certain properties, other terms may be grouped together.

According to G. Salton (SAL,68) the main steps in the semi-automatic thesaurus construction can be summarised as follows.

1. A word frequency list is prepared (usually by automatic methods).
2. The different word usages for each word to be included

in the thesaurus are decided upon.

3. Questions are prepared which serve as a means by which term grouping can be done.

4. These property matrices are compared and words, identified by like properties, are assigned to the same thesaurus category.

2.4 AUTOMATIC CONSTRUCTION

G. Salton worked extensively (SAL,68) on automatic thesaurus construction. He states that when automatically generated keywords are compared with terms manually assigned by subject experts, it is usually found that there is a 60% to 80% agreement (SAL,73A). He goes on to compare the effectiveness of fully automatic text processing methods with manual retrieval operations using his experimental SMART system and the MEDLARS retrieval system operating at the National Library of Medicine in Washington, D. C. The SMART system operated without any manual content analysis, whereas the MEDLAR was using conventional methodologies of trained subject experts who were assigning keywords to all incoming documents, and using a printed thesaurus known as MESH (Medical Subject Headings). Using different automatic retrieval methodologies on the SMART system, on page 275, he concludes that: "...the SMART relative recall is about 25% better on the average than the MEDLARS average recall. The improvement reaches 80%

to 90% on the average for the more sophisticated SMART methods such as correlation cut-off and feedback searches, and the differences then are statistically significant",

These methodologies and the concept of recall will be dealt with in later chapters.

For some years, experiments were conducted to an automatic determination of thesaurus classes based on the properties of the available document collections. The general process is described by Salton (SAL,72A).

1. A term document matrix is constructed,
2. from the term document matrix, a term-term similarity matrix is generated by computing the similarity between each term vector.
3. A binary term-term connection matrix is developed by applying a threshold value to the term-term similarity matrix. Two terms are assumed to be connected whenever the similarity between corresponding term vectors are sufficiently high.
4. This binary connection matrix can be assumed as an abstract graph and subgraphs of this graph can be used to define classes of terms or clusters.

The term-term connection matrix is, in general, a large square matrix of integers. They may be used with one of the well known cluster algorithms (SPAT,80) (KAZ,80) to group the

terms into clusters. Such clusters may be unstructured sets of terms or may be arranged into a hierarchy of clusters. The cluster analysis algorithms fall into two groups:

1. divisive methods that produce clusters by splitting up a large group into smaller groups proceeding in a top-down manner;
2. agglomerative methods that combine smaller clusters into larger ones proceeding in a bottom-up manner.

In practical application of thesaurus construction procedures, one faces several thousand terms and consequently the matrix size is very large. Care must be taken in the choice of a clustering algorithm to make the approach computationally viable. The creation of the binary term-term connection matrix is time consuming and therefore expensive for a large size term collection. Using similar procedures, as described here, a number of investigators have constructed automatic term classification models.

Dattola describes a fast algorithm for automatic classification (DAT,68A) where an existing classification is improved by a selective modification of the original classes. The basis of his algorithm is the partitioning of a document collection into equal size clusters. Associated with each set of documents, there is a corresponding profile vector, which consists of the ranking values of all the concepts from the document set of the cluster. The concepts are ranked in decreasing order of the number of documents in the cluster. A

constant (or base value) minus this rank of the concepts will constitute the rank values in the profile vector. He also acknowledges the fact that a fairly large collection of documents is not feasible to classify with most automatic procedures and he is proposing his method which can classify hundreds of thousands of items into useful clusters in a reasonable amount of time.

A similar approach is described by Hoyle (HOY,73). Using a properly chosen sample or a categorized collection of documents, the probabilities of the presence of a specific word in each of the categories are calculated. The words are then listed with their probability in each category to form category lists. To index a document, its words are tried against the words of each of the category lists and the word weights for the matches are summed within each category. The document belongs to the category whose match weight is the largest. The word matches and weights for the selected category are kept as the keywords of the document. Using this automatic methodology, 97 of the 124 documents (78%) were in agreement with professional indexers.

2.5 CONCLUSION

Over the years, Salton et. al., have shown that when automatic and manual indexing techniques are compared, one normally finds a fairly high degree of agreement. Not only time

is saved by employing automatic or semi-automatic thesaurus construction methods, but they eliminate the unconscious bias of individuals in choosing thesaurus and associated relationship entries. The vocabulary bias of the authors of the representative document collection, however, plays an important role in the future contents of the thesaurus. This bias can be overcome by careful selection of the document base.

CHAPTER III

INFORMATION STORAGE AND RETRIEVAL

Thesauri are used with automatic libraries and on-line systems. The role of these systems is to provide access to a variety of bibliographic files. In 1977, there were two million bibliographic references available for on-line searching (ANN,78). How do we cope with this information explosion? How much information is relevant to a query? In this chapter, we present a description of a generalized information retrieval system with its component parts and feedback mechanism, the possible file structures and access strategies used, the measure of relevancy, and types of information retrievals.

3.1 FILE STRUCTURES AND ACCESS STRATEGIES

3.1.1 INTRODUCTION

Information systems typically handle a very large volume of data while computations on each piece of data are generally simple. This is in contrast to those scientific applications where complicated and prolonged mathematical and logical operations are performed on relatively small amount of data. At the present state of computer technology, it is not possible to

store in the main memory of a computer even a small portion of the data base used in an information system. Data are therefore placed on backup storage devices such as disks, drums, tapes, and small portions of the stored data are brought to the main memory during the processing task. Since the speed of data transfer between backup storage and main memory is several orders of magnitude slower than the computational speed of the central processing units, it is important to transfer as little data as possible. Therefore the goal of file structure designs is to construct access paths through records in a data base in order to reduce the amount of data transfer for each task.

In many information systems, a data record is represented by a fixed set of attribute-value pairs where each attribute describes a certain characteristic of the object that is represented by this record. If a data base administrator decides to use six attributes (e.g. Social Insurance Number, Name, Department, Job, Salary, Education) to describe employees, every employee record would contain exactly six attribute-value pairs. Such a fixed set of attributes presumably would capture all the properties that one wanted to talk about. We call this kind of file an ATTRIBUTE-BASED file. Many data base systems are designed upon this premise.

In some situations, however, the properties of objects may not be fully described by a predetermined, fixed set of attributes. The library document file is a good example.

Besides attributes such as Author, Date of publication, Publisher, etc., one often includes a set of keywords that are extracted from the document to further describe the document content. We will call such files KEYWORD-AUGMENTED files. A record in this case would be more general than a record of an Attribute-based file; it would contain not only a fixed set of attribute-value pairs but also a list of keywords. Such inherent differences between these two types of files need not mean that they cannot be maintained by some common file structures. In fact, many file structures are applicable to both type of files and formally we can consider a keyword in a file as an attribute, calling such attributes keyword attributes. Each record in the file then "conceptually" consists of a set of predetermined, fixed attributes plus a large number of keyword attributes. Within this conceptual framework, we have a unified view for these two types of files and in the rest of the chapter, a file structure is assumed to be applicable to both types of files unless it is otherwise mentioned.

As it was stated before, the objective of this thesis is query enhancement, which is meaningful in an interactive system with its quick response. Here the user is able to judge the effect of enhancement on his query. One of the controlling factors of quick response is file structure. The following file structures and/or their combinations are commonly used.

3.1.2 EXACT FILE STRUCTURES

SEQUENTIAL FILE. There is no organization per se in a sequential file. Records are stored one after the other in the order in which they were submitted to the system. The addition or deletion of records is a simple process, but the retrieval of a specific record requires a time consuming linear scan.

LIST STRUCTURED FILES. In this technique a "pointer" is included in each record. It "points" to the location of another record in the file, thus it is called "linked list file". This pointer technique allows the logical and physical arrangements of records to be different.

Alterations to a file are more efficiently accomplished by the use of the linked file than the linearly ordered record file. The disadvantage of a linked file is noticeable when searching for records. There is only one record which is accessible from any other record and hence a chained or linked file must be searched serially. Records with one pointer allow a forward or backward traversing of the file, while records with two pointers give the flexibility of forward and backward search of the file.

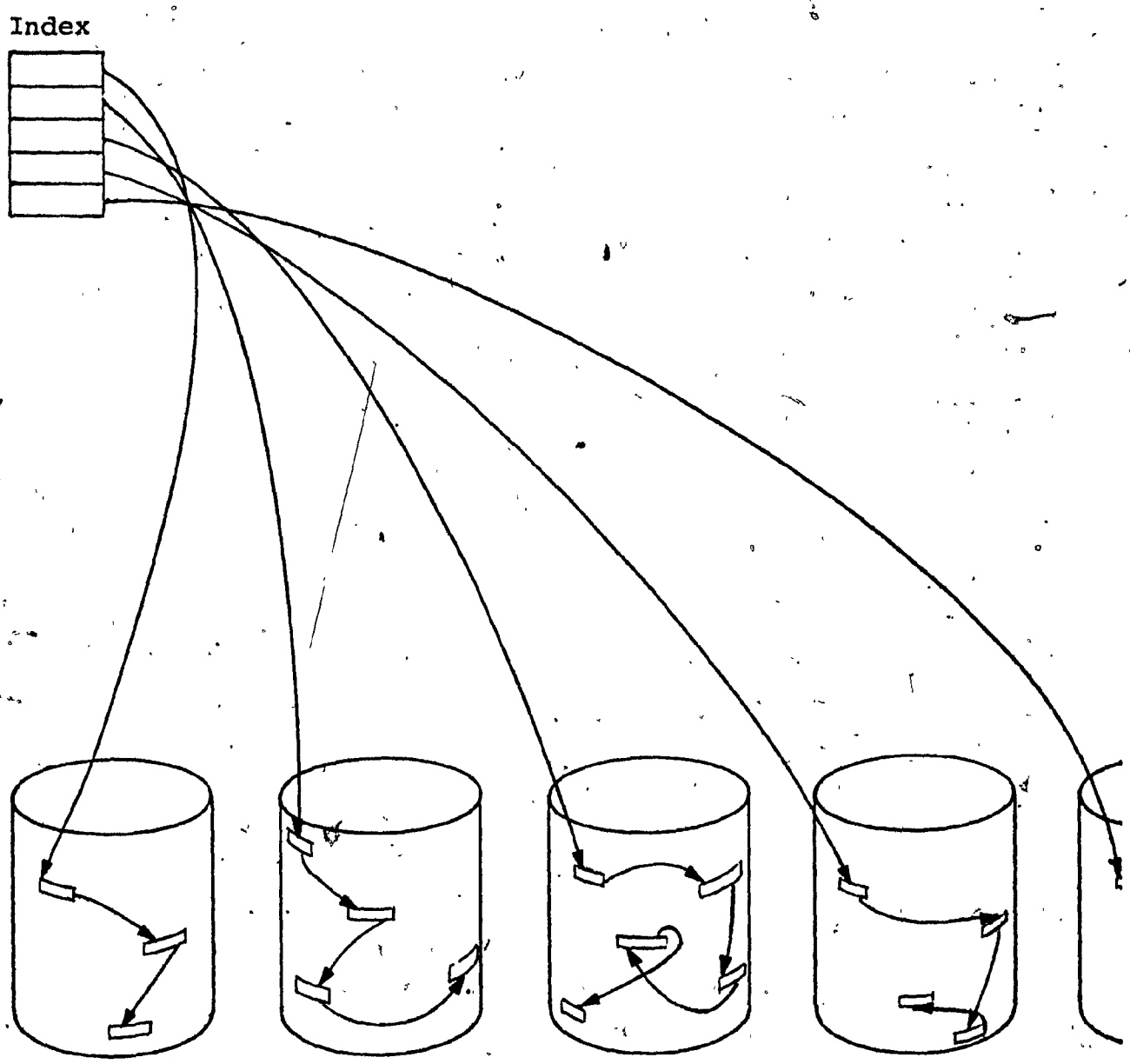
If the items or records are chained together in sequence, several techniques can be used for shortening the chain search time (MAR,75).

MULTILIST CHAIN. When the chain is divided into segments, and an index gives the value of the first item in each segment with its pointer to the first record, it is known as a multilist chain.

CELLULAR CHAIN. The multilist chain can be organized so that no part of it extends beyond a certain hardware cell or boundary selected, thus minimizing access time. For example, a cell can be confined to the size of a disk track so that each segment is in core when it is searched.

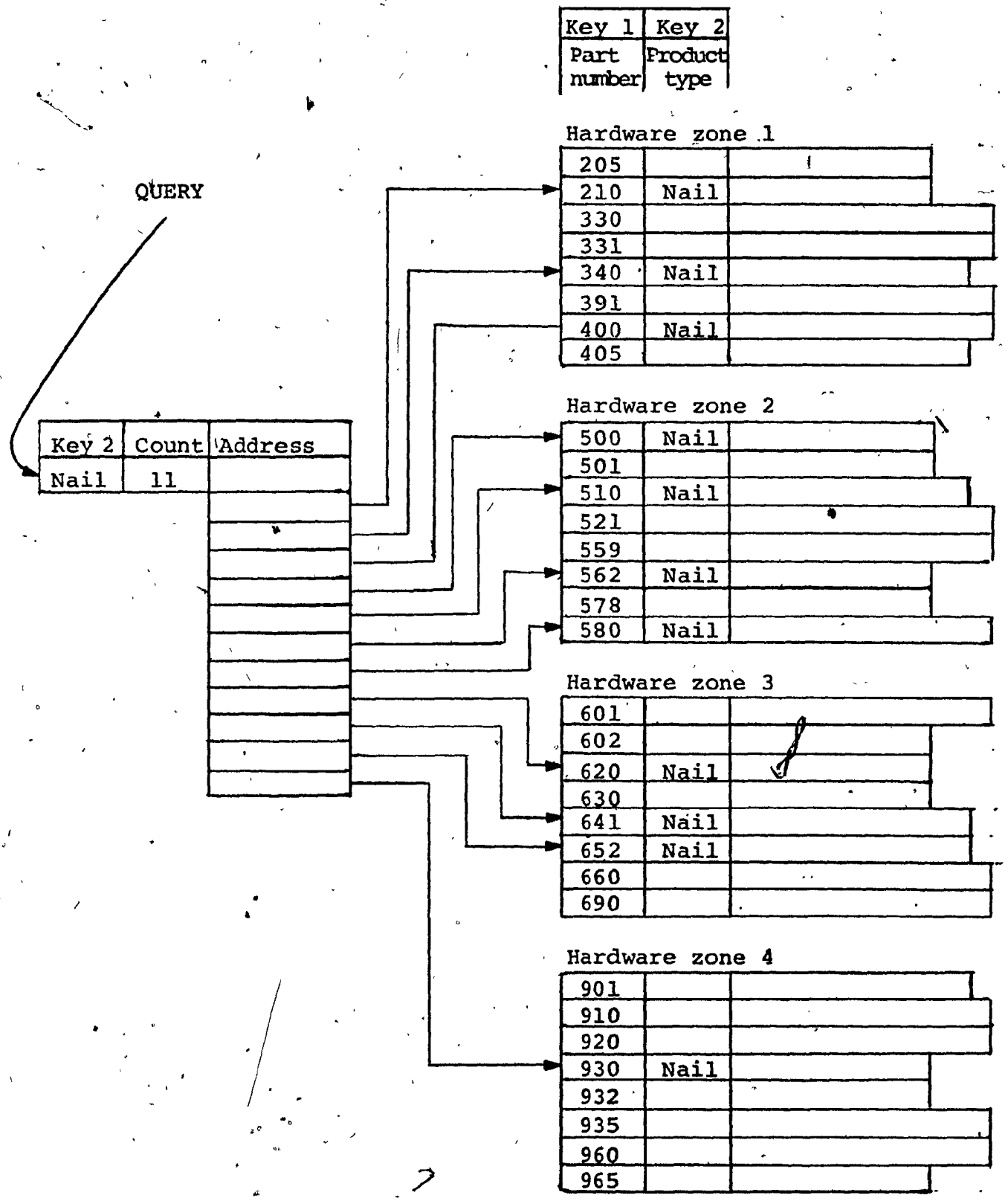
PARALLEL CELLULAR CHAIN. Cellular chains may be organized in such a way that they can be searched in parallel. The segments are spread across modules which can be read simultaneously. Figure 3.1 illustrates this file organization.

INVERTED LIST. For a multilist file the number of "pointers" or record addresses can vary in an index from one address pointing to the head of the list, to one address for each item. If there is an entry for every record in the index, the organization is referred to as an inverted list. The inverted list organization gives the fastest response to real-time inquiries because no chain have to be followed. On the other hand the indices can become enormous, and the organization of the indices themselves becomes a major file problem (see figure 3.2).



PARALLEL CELLULAR CHAIN

FIGURE 3.1



INVERTED LIST FILE

FIGURE 3.2

CELLULAR INVERTED LISTS. The size of the index file can be reduced by storing, not the address of each record, but an indication of a hardware zone, say a disk track, in which it resides. The zone will then be searched to find the requested record.

PARALLEL CELLULAR INVERTED LIST. As in the cellular chain organization, the cell position may be spread across several disk packs so that the cells can be searched in parallel in order to minimize system response time.

TREE STRUCTURES. Knuth (KNU,73) defines a tree as follows: ".. a finite set T of one or more nodes such that,

1. there is one specially designated node called root of the tree.
2. The remaining nodes are partitioned into m>0 disjoint (i. e. not connected) sets T1,...Tm and each of these sets in turn is a tree. The tree T1,...Tm are called the subtrees of the root".

Trees are used in both logical and physical data descriptions. In logical data descriptions, they are used to describe relations between segment types or record types. In physical data organizations, they are used to describe sets of pointers and relations between entries in indices.

E. H. Sussenguth proposed to use the tree structure to store English words. He states (SUS,63) that: "The filial set of each Kth level node is comprised of those nodes

which correspond to those elements actually used in combination with the element associated with the parent node".

See figure 3.3 for an application of this concept. The filial set of the letter B, for example, would be the letters that can be used with B to start a word. A partial key can be associated with each node. The key value is made up of the node values from the root to the given node, which means that the key for a leaf is the key of the record to which the leaf corresponds. In the example given in figure 3.3, BAN is a partial key with respect to BAND and BANE and it is also a record key, therefore it is a leaf. By terminating each key with a special character (* in figure 3.3), records of the file correspohd to leaves.

One possible representation of this technique is to chain all nodes to their filial sets (heir pointer) and to chain the nodes within the filial set (twin pointers) together. This technique is called "double chaining" and, in essence, it creates a binary tree representation of the file. In terms of computer storage, one portion of a word might contain the address of another node on the same filial set level (twin pointer) and a second portion might contain the address of the first node of the filial set one level lower (heir pointer). The actual text of the record might be stored in a third portion of the word, or it could be "pointed to", in which case the file

would be "triple chained". Figure 3.4 gives the computer memory contents for the tree of figure 3.3. The chaining technique can make use of any available memory location for adding information about new file additions.

3.1.3 APPROXIMATE FILE STRUCTURES

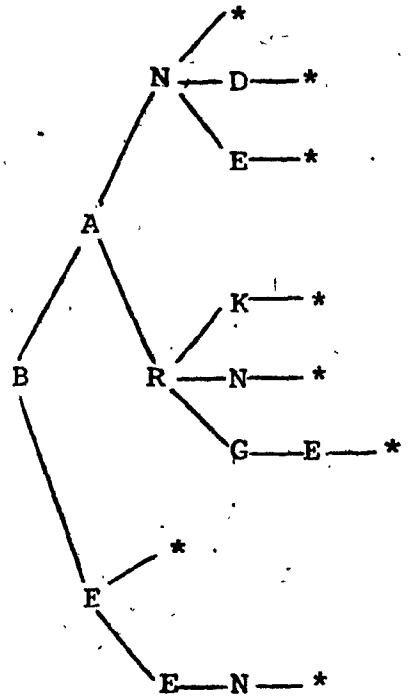
One of the most important aspects of any information retrieval system is the time: how quickly a user's request can be processed, the specific information generated and the output returned to the user. For a large sized document collection, search time - the time spent scanning and correlating against the members of the collection - is critical. It can become excessive because it often varies with the size of the collection. Because of this, various techniques have been developed to shorten search time.

3.1.3.1 CLUSTER TREE ORGANIZATION

Clustering is an operation which divides a document collection or document space into several groups, each of which is considered as a unit. Each cluster is represented by a "centroid", similar in form to the documents it represents. Many clustering algorithms have been used in experimental systems such as developed by Bonner, Rocchio, Salton, Dattola and C. T. Yu (BON,64), (ROC,66), (SAL,67), (DAT,68), (YU,74). Most of these systems make use of correlations between the

BAN
BAND
BANE
BARK
BARN
BARGE
BE
BEEN

File contents



Tree representation

FILE AND ASSOCIATED TREE STUCTURE

FIGURE 3.3

1	B	-	2	8	E	-	11	15	G	-	20
2	A	3	4	9	K	13	14	16	*	-	-
3	E	-	5	10	D	17	18	17	E	-	19
4	N	6	7	11	N	-	12	18	*	-	-
5	*	8	-	12	*	-	-	19	*	-	-
6	R	-	9	13	N	15	16	20	E	-	21
7	*	10	-	14	*	-	-	21	*	-	-

MEMORY MAP OF TREE IN FIGURE 3.3

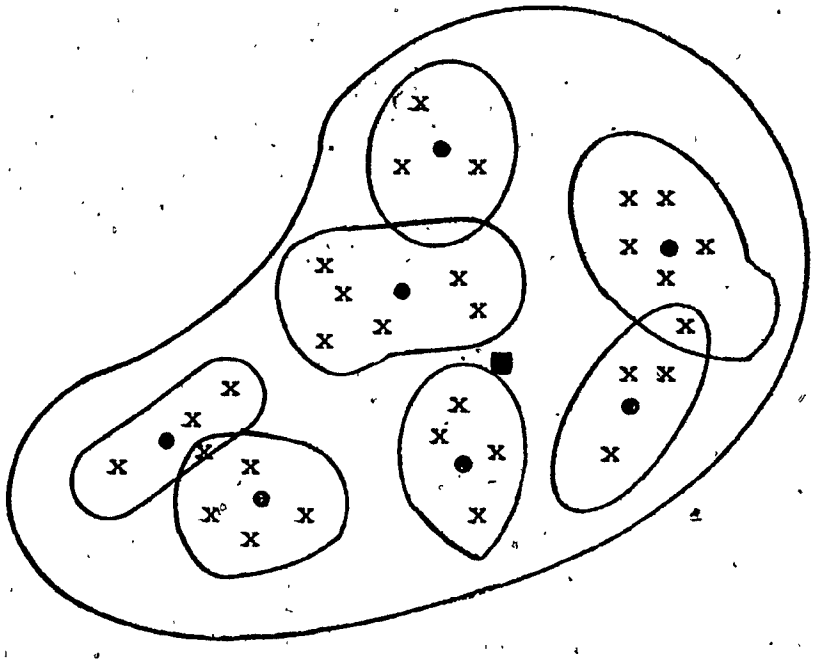
FIGURE 3.4

documents to be clustered. For Salton's SMART system, each cluster is identified by a representative cluster profile, somewhat akin to the center of gravity of a set of mass points. This cluster profile is a set of weighted terms, representative of the documents included in the corresponding cluster (SAL,72B).

Bonner's technique is to construct a similarity matrix, which specifies the closeness of each document with respect to every other document. Some information is extracted from the matrix to determine the cluster. One way is to represent the documents as vertices of a graph whose edges are determined by the values of the similarity matrix. There is an edge between the ith and the jth vertices if, and only if, the ith and the jth documents are "sufficiently close". The clusters are defined in terms of graph theory. In general, each cluster is represented by a vector which is a mathematical combination of the documents it represents. Finally the vector is represented in a balanced tree organization for a generally excellent retrieval performance (figure 3.5).

3.1.3.2 NEAR - NEIGHBOR SEARCHING FILE STRUCTURE

For multi-key or associative searches Bentley (BEN,75) proposes a new type of data structure, called multidimensional binary search tree or k-d tree. A k-d tree is defined as a file, whose records are stored as nodes in the tree. In



● Cluster centroid

■ Main centroid

DOCUMENT CLUSTERS

FIGURE 3.5

addition to the k-keys which comprise the record, each node contains two pointers pointing to another node in the k-d tree. The k can be considered as the dimensionality of the records, and each pointer as specifying a subtree. The relative magnitude of the keys and the order in which the records arrive are relevant only for the construction of a k-d tree. Using a k-tuple permutation, the nodes are randomized. The first node, say P, in the collection will become the root. This includes a partition of the remaining nodes into two subcollections. If two new records, say Q and R, fall in the right and the left subtrees of P respectively, their relevant ordering (that is, whether or not Q precedes R) in the original collection is unimportant.

The k-d tree can be used for both intersection queries and best-match queries. Intersection queries specify that the records to be retrieved are those that intersect some subset of the set of valid records. The simple query or "point search" (FIN,74) asks if a specific record is in the data structure. The next more complex intersection query is one in which values are specified for a proper subset of the keys. If values are specified for "t" keys, where t<k, then the query is called a "partial match query with t keys specified". The most general type of intersection query is one in which any region at all may be specified as the set with which the records to be retrieved must intersect, hence it is called "region query".

The best-matched query, generally termed as near-neighbor searching, normally deals with the search of all records within a fixed distance "r" of the given query. Given "n" records, it finds all pairs of records that are within the distance r of each other, or it finds the "k" most closely matched records for a query.

This class of file structure is applicable to an environment where the records can be considered as points in a multidimensional metric space and a query is a point in this space. For example, locations of warehouses can be specified by their latitudes and longitudes; a transport company may want to find the warehouse that is closest to a given point. The idea of distance can, in this case be measured by the Euclidian distance. There is a large class of such near-neighbor searching problems; however, many information storage and retrieval problems do not fit into this category.

3.1.4 FILE COMPRESSION

One research area of great concern is that of handling (i.e. generating, updating, searching) large data bases. Some of the major data bases are added to at the rate of half a billion characters of data per year. Searching a data base of this size over a five year span could become a problem. Several organizations have done research on compression of bibliographic data bases.

A generally good way to compress data for storage is to use the most efficient form of character encoding. The conventional 8 bit EBCDIC code which most files use does not give tight encoding for data, nor does the ASCII code which is widely used for data transmission. Alphabetic data and most punctuation could be stored in 5 bit characters, as is the case of the BAUDOT code used for telegraph transmission.

A tighter packing of data can be achieved with a code which employs a variable number of bits per character: With such a code, the most commonly occurring characters would be short, and the infrequently occurring characters would be long. This type of coding was originally proposed by D. A. Huffman and it is called after him a Huffman code (HUF,52).

In general, the techniques used recode the data in a more compact form. For example, Heaps & Thiel (HEA,70) have compressed text tapes by assigning a two-byte code to each term on the tape. For coding and decoding, tables are required. They feel that in order to design a large data base for retrospective search - the term will be explained later in this chapter - there should be automatic data compression, a minimization of the number of input/output operations to the direct access files, and a minimization of the requirement for internal core memory (THI,72).

A variable length encoding described by Wells (WEL,72) obtains a compression of about 20% for free format files. He

states that the economic effect is not so much in the saving of storage as in the reduced traffic to and from storage devices.

3.1.5 ACCESS STRATEGIES

In many applications, files are both searched frequently for pertinent information and altered frequently with new information, such as an airline reservation system. In other applications the files are subjected to the following situations.

1. Accessed frequently and altered infrequently, such as an on-line query system.
2. Accessed infrequently and altered frequently, such as a data base storing telemetry data.

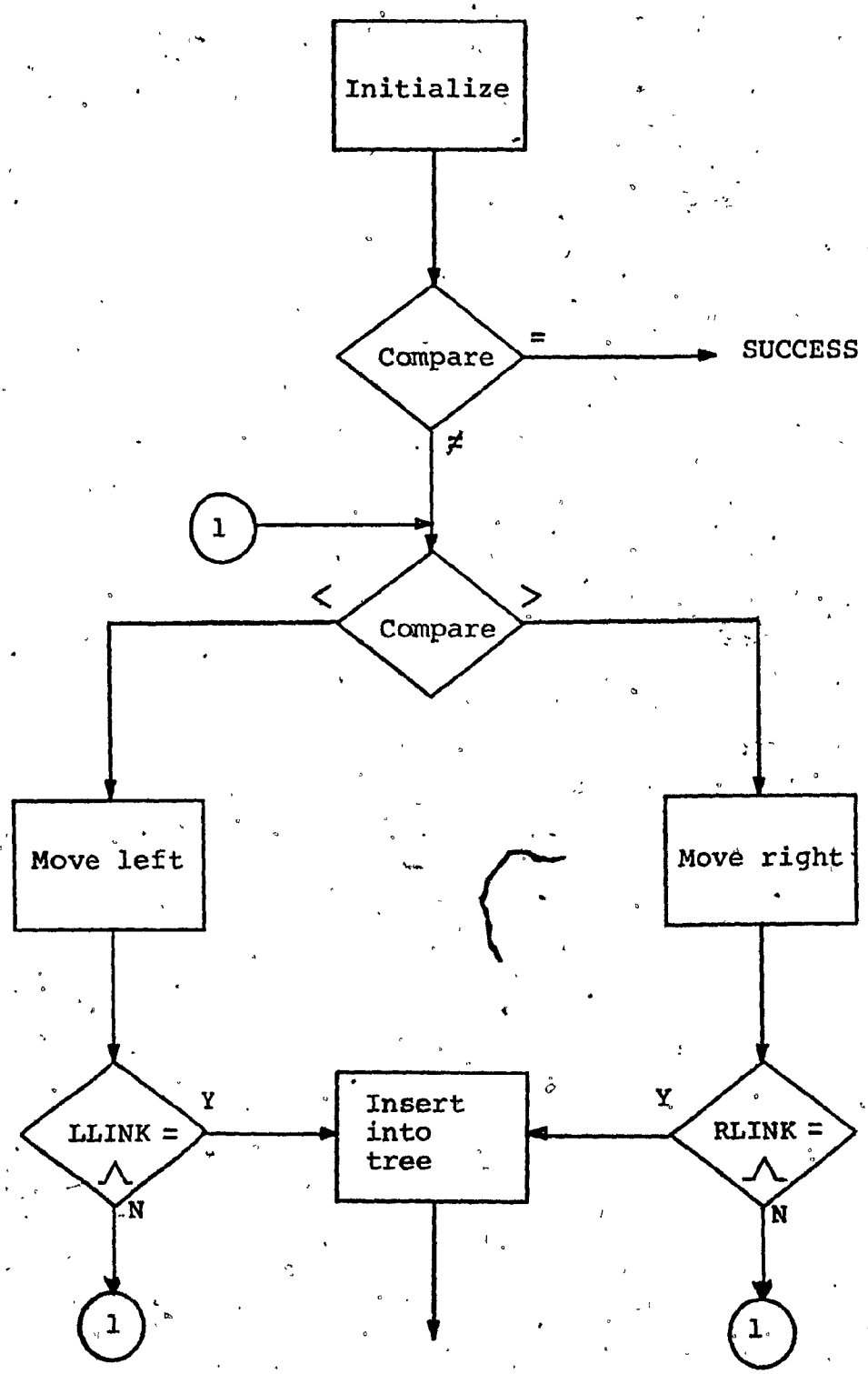
In any thesaurus file, the information is regarded as being both frequently accessed and frequently altered. However, the accession feature is of greater importance than the alteration feature, because once the thesaurus has been created, information is more likely to be accessed than altered. The access strategies discussed on the following pages will fall into one of the three application characteristics of a file. Although some of the access methods are too well known to be described in any detail, for the sake of completeness, it was decided that they be listed also, without much description, among the lesser known strategies.

SEQUENTIAL SEARCH. This is the simplest and perhaps the crudest way of locating a record and it is only likely to be used on a batch processing operation using a serial file, such as tape, in which each record must be read anyway.

BINARY SEARCH. This technique is an effective method of handling files which are searched frequently and altered infrequently. The alteration of the file is time consuming because the records in the file must be in an ascending or descending order of key and many records might have to be moved either to make room for inserting a new record or deleting an existing one.

PARTIAL KEY SEARCH OF A TREE STRUCTURE. The searching of tree allocation for a given record or item is a simple procedure. Initially the roots are scanned to find the root corresponding to the first element of the key of the wanted record. After the root is located the "children" or the "filial set" of the root is accessed. This filial set is searched for the second element of the key of the required record. Searching the filial sets for the elements making up the key continues until the required leaf is located (see figure 3.6).

Basically the same procedure is being used to add a record or element to the file. The file is entered as if a search were being undertaken. At some level a filial set is found that does not contain a node value that matches one of the elements of the



TREE SEARCH AND INSERTION

FIGURE 3.6

key being added. At this point the filial set is expanded by including as a node the element of the key which previously was not a member of the filial set. The filial set of this newly added node consists of the next element of the key being added to the file. Additional filial sets are added until all elements of the key are added to the file.

HASHING. The search methods discussed above were based on comparing a given key "K" to the keys in a table. Another methodology is to do some arithmetic calculation on the key, "f(K)", converting it to an actual address "H", which can be used to store the key and its associated data in an area of the storage medium. This technique is called hashing and it is superior to binary search and tree search from the standpoint of both speed and space, except that binary search uses slightly less space (MAR,75).

3.2 FULL TEXT SEARCH

While bibliographic data bases are still the major sources of information retrieval systems, there is a growing interest in actual information files. Full text data bases are being used in law retrieval systems such as WESTLAW of the West Publishing Company, which has provided indices to State and Federal court cases. Scientific literature is becoming available in full text form both in the field of Chemistry and Physics. These files are seen as the next step in the on-line revolution, providing

answers to questions rather than references to the literature where such answers may be found.

Up to now we discussed files which contained fixed attributes and a number of keywords (see paragraph 3.1.1). In this section we discuss files, which contain the full text of articles and documents.

Full text search is usually based on search words which are posed against the text. Usually the "distance" or the number of words between the search words can be regulated. The parameters defining this distance have a direct bearing on the relevance of the text retrieved. For example an "adjacency" of 10 words usually will yield a more general text than when the "adjacency" between search words is defined as 1.

O'Connor uses a novel concept of "syntactic joints" as a search criterion (OCO,73). A syntactic joint is a conjunction, preposition, or punctuation mark. The search score for a sentence is based on the number of intervening syntactical joints as well as intervening words between clue words, with a higher ranking given to word sets having between joints. One or more sentences, "answer passages", are retrieved for review by the user, instead of the usual citation information, with pairs of sentences selected based on "connector" words such as "therefore", "thus", and other indicators of links to other sentences.

In the future, scientific literature may become available in full text form as more journals convert to computerized photocomposition. The American Chemical Society and the American Institute of Physics are in the process of conversion now, but at the moment the field of law is the leader in full text retrieval. There is a great need and an obvious payoff for having the verbatim transaction of a law, regulation or court decision. LEXIS, the service of Mead Data Center, is the oldest and most used commercial data base service in law and it provides access to a vast store of legal texts.

3.3 MEASUREMENT OF RELEVANCE IN INFORMATION SYSTEMS

3.3.1 EXACT AND APPROXIMATE RETRIEVAL

According to the way in which a retrieval criterion is written and interpreted, we can talk about exact and approximate retrieval. In an exact retrieval query, the retrieval criterion is normally a Boolean and/or arithmetic combination of the keys. For example, a query that requests all records in the Telephone Company for employees whose salaries are higher than the Company average, is an exact retrieval query. We call these queries exact because an arbitrary record in a file either satisfies a query precisely or it does not satisfy a query at all. This is a 0-1 decision on relevance.

In some other retrieval situation e.g., document retrieval systems, pattern match systems, etc., one does not have clear cut decisions. One will have to talk about the degree to which a record is relevant to a query. The SMART retrieval system (SAL,73A) is such an example. A query might consist of a set of keywords and the degree of relevance between one record and one query would be a function of keyword, matches and keyword mismatches. These queries might also include additional Boolean and/or arithmetic combinations of keys. For example, a query that retrieves "documents that are published after 1973 and that contain as many of the three keywords: ARTIFICIAL, INTELLIGENCE, SIMULATION as possible" is a mixture of an exact criterion and

an approximate criterion. We classify these queries as approximate retrieval queries.

3.3.2 PRECISION AND RECALL

When a user describes his interest to the retrieval system by means of a question (query), it initiates a search of the data base. Ideally, the list of data base items he receives should contain all the relevant, and only the relevant, items to his interest. The extent to which both criteria may be met depends on whether his question adequately represents his interest and also whether the terms used in his question are terms used to describe, or index, items in the data base.

RECALL may be defined as the proportion of relevant material actually retrieved. PRECISION is the proportion of retrieved material which is actually relevant. The following examples will illustrate the two concepts.

Suppose that a user of a retrieval system receives a list of "M" items in response to a search and, after examining the items, he decides that "R" of them are relevant to his interest; the search is then said to have a "precision" or "relevance" equal to R/M. The precision is equal to unity only if every retrieved item is relevant to the user's interest.

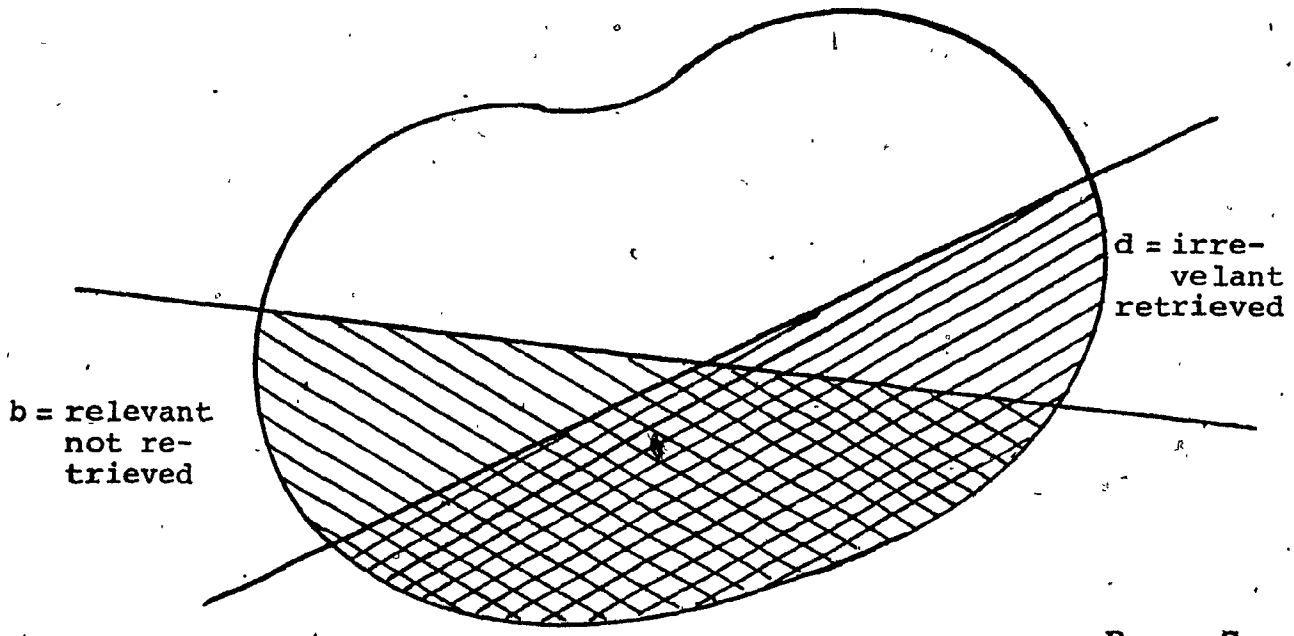
$$\text{PRECISION} = R/M$$

Suppose now that, after receiving the list of M items of which R are relevant, the user manually examines the entire data base and finds that it contains a total of S relevant items in the collection; the search that produced the list of M items is then said to have a "recall" equal to R/S. The recall is equal to unity only if the search located all the relevant items.

$$\text{RECALL} = R/S$$

Salton (SAL,68) separates the document collection into four parts: retrieved and not retrieved documents and documents that are relevant and non-relevant. This partitioning of the collection is illustrated in figure 3.7.

The variation of precision and recall, as a question is made more specific or more general, is shown in figure 3.8. The point A corresponds to a question which is formulated so that all retrieved items (probably very few) are relevant, thus R/M is approximately unity, and R/S is approximately zero. As the question is successively generalized, the precision ratio drops and the recall ratio increases; these changes are represented by the points B, C, D, E. The point F corresponds to a question which is designed to retrieve almost all the relevant items at the expense of including many that are non-relevant. In this case the R/M ratio is close to zero and the R/S ratio is approximately unity.

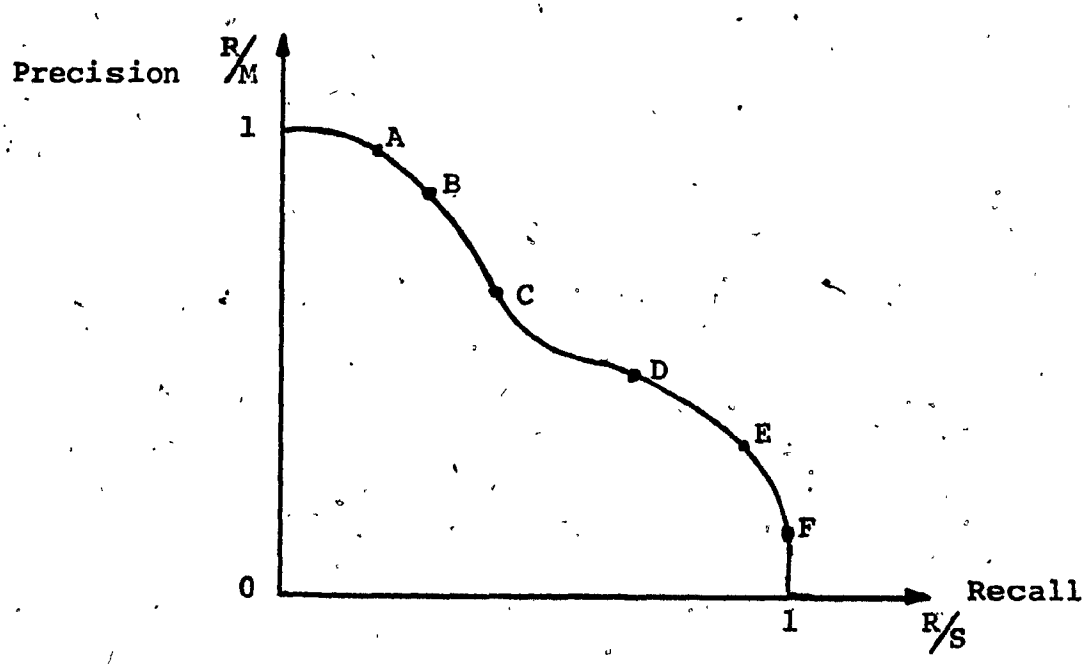


$$R = \frac{c}{b + c}$$

$$P = \frac{c}{c + d}$$

PARTITIONING DOCUMENT COLLECTION

FIGURE 3.7



CHANGE IN PRECISION AND RECALL

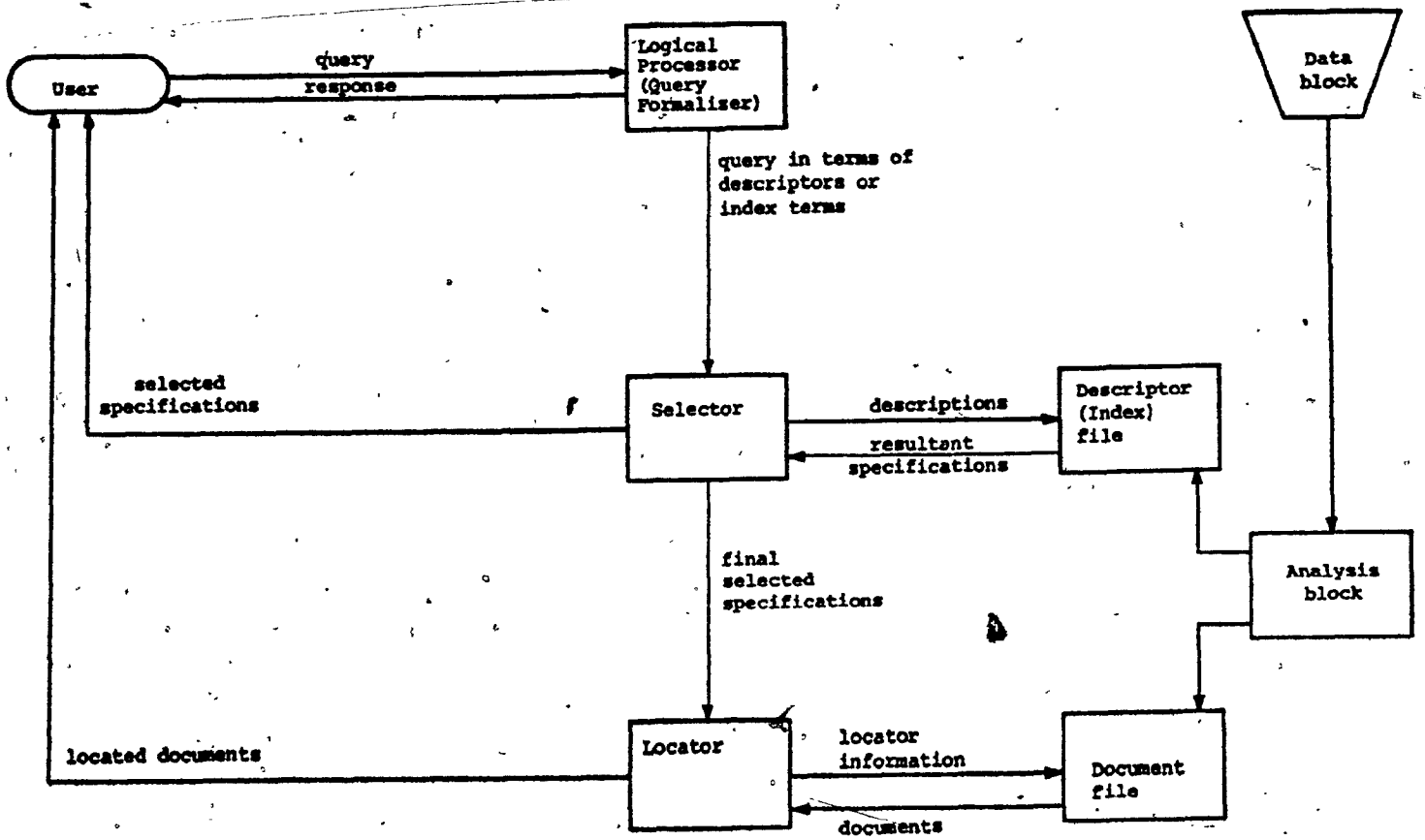
FIGURE 3.8

3.4 MODEL OF AN INFORMATION STORAGE AND RETRIEVAL SYSTEM

A generalized model of Information Storage and Retrieval System (ISR system) was proposed by C. Crouch (CRU,71). The phrase "information storage and retrieval system" is defined as a system which retrieves documents or document references in response to a specific request. Thus the system, regardless of whether it retrieves complete document texts, document surrogates (such as abstracts or extracts), or the names and addresses of documents, is termed an information storage and retrieval system.

For a graphical representation of the generalized ISR system, see figure 3.9 (after Crouch). The sequence of operations performed follows the flow from the User to Logical Processor, Selector, Descriptor file, Locator, Document file, and back to the User. In addition to the User, another point of input to the system is through the Data block. This information is then processed by the Analysis block.

The user inputs his query to the system. Using a so-called "man-machine" dialogue with the ISR system, it allows the user to modify his original query to facilitate retrieval of the desired information. This "man-machine coupling" is generally needed so that each can contribute to the best of his ability at each step in the way. For example, the system might help the user in formulating the request by noting with each change the



MODEL OF THE GENERALIZED INFORMATION STORAGE AND RETRIEVAL SYSTEM

FIGURE 3.9

probable number of documents in the final answer, by presenting representative documents for evaluation, and by ranking the output according to degree of relatedness. The user, on the other hand, could help the system find the required answer by correcting possible misunderstandings of the request as early in the search as possible, by narrowing or broadening the request if the size of the expected answer becomes too large or too small, and by continually refining the request based on the information supplied by the system.

The Logical Processor, or Query Formalizer takes the input query, reducing and formalizing it to a form recognizable to the ISR system. This reduced query, in the form of the system's descriptor language, is then output to the Selector. The Logical Processor may be responsible for pre-search activities, which are based on term and dictionary displays of previously stored information. The user examining this information can decide how to reformulate his query to obtain the best result. Salton (SAL,68A) lists the following types of pre-search information.

1. Lists of terms included in the user's original search formulation together with word frequency information giving the frequency of use of each word in one or more of the stored document collections.
2. Thesaurus excerpts corresponding to the terms included in the user's search formulation, and consisting, for each of the originally available terms, of a complete

thesaurus class, including synonyms and other terms related to the original.

3. Title and abstract of source documents, that is, of documents originally known to the user as relevant to his search query.

The query enhancement of this thesis belongs to the pre-search activities of the Logical Processor.

The Selector uses the formalized query to search the Descriptor (or Index) file. Using the expanded query as input, it retrieves from the Descriptor file the set of all documents that are associated with each descriptor in the expanded query and it performs the indicated operations upon these sets according to defined priorities. The result is a final set of specifications, i.e., pointers to all those documents which have been found to be associated with the query. These specifications are then passed to the Locator.

The Selector may also interact with the user by means of user feedback. This relationship is indicated in the model by the arrow labelled "selected specifications". The selected specifications might be a list of document identifiers associated with a particular query term. Such functions are called post-search activities. In post-search activities, the Selector could show the user the result of an initial search. One application of the method would be the user's reformulation of his search request based on the data provided by the system.

Another application would be the automatic query reformulation, which is using relevance feedback. In relevance feedback, the user is given a set of items retrieved using his original query. He is then asked to judge which items of this set are relevant to his request. This information then is used to automatically generate a new query for another search. This feedback process can be iterated as often as desired.

The Descriptor file contains all index terms and, generally, it is a large, stable file whose entries are by nature alphabetic and of variable length. The most important factor in an ISR system is fast response time and for the Descriptor file the preferred file organization seems to be the inverted file or some variation thereof, as were discussed in Section 3.1.

The Locator searches the Document file to extract the document information associated with each document for the set passed to it by the Selector. Accepting as input the set of documents determined by the ISR system to be associated with the query, the Locator retrieves the entry associated with each document from the Document file. The entry may consist of the document title, an abstract, or an extract.

The Document file is composed of entries which are the ISR system's representation of the corresponding documents. The documents of the file are dependent upon the original form of the document as it enters the system through the Data block as

well as upon the operations performed upon it by the Analysis block. The final document representation is largely determined by what is considered adequate or necessary in a document surrogate for a particular system.

The information contained in the Descriptor and Document files enters the system as raw material via the Data block. This information consists of unmodified textual information in whatever form is found to be convenient. For a generalized document retrieval system, each informational set is considered as a set of characters only, i.e., a set of recorded symbols which are recognizable within the confines of the particular system.

The main function of the Analysis block is to extract from the incoming data two different kinds of output.

1. Some indication of the content of the incoming document to be stored in the Descriptor file along with a pointer to the document in the Document file. Since a document is represented by its descriptors, they must be indicative of its content; they are in fact clues to the document content.
2. A representation of the document itself (i.e., the system's representation of the document), to be stored in the Document file.

As can be seen, the Analysis block is responsible for restructuring, transforming its input into two constituent

parts, each of which serves as input to another component of the system. For very large document collection, the search of an entire file is prohibitive in terms of time and processing cost, thus it is divided into clusters (see Section 3.1.3.1) for faster and more effective search. The Analysis block has this subsidiary function to utilize a clustering technique in order to reduce the problem somewhat, allowing it to be handled economically within the ISR system.

3.5 INFORMATION SYSTEMS

The information stored in a data base can be classified in general as historical or retrospective information and current information. The length of time it takes for current information to "age" to be historical, is solely dependent on the type of application. In satellite tracking, the telemetric data of two hours is historical, whereas in a bibliographic data base, an article in a monthly periodical can still be considered as current after three weeks.

A number of document data bases are created in the form of regular issues of magnetic tapes that contain references to the most recent articles published within some set of journals (SCH,73). For example each issue of a tape created at bi-weekly or monthly interval might contain 6000 references to articles that have appeared in some 700 journals. The user of the search service requests a list of all the articles that contain certain

combinations of title words, author names, subject headings and so forth. His specified set of terms is considered to form his "interest profile". Some of the users might have standing interest profiles to be searched on each issue of the tape as it appears. The service to provide such regular searches with respect to standing profiles is called Selective Dissemination of Information (SDI) for current awareness. In contrast, a search on all back issues of a document data base is considered historical or "Retrospective Search".

SDI services provide for the running of searches against only the new material being input to a bibliographic data base. Its popularity is not surprising because such searches are much less expensive than the larger retrospective search. Many commercially-run information services, or Information Dissemination Centers, provide for the users to set up their own individually tailored profiles. These profiles usually feature capabilities for logic, term truncation, weighting, nesting and search on various types of terms or data element types. Besides the individually tailored profiles, many SDI centers also provide standard profiles or group profiles at a lower price than individual profile cost.

The conventional retrospective search can be done both on-line and in batch mode. In general, on-line search is preferred by most users. Most of the tape driven batch systems are being replaced by the more convenient, direct access device

driven on-line systems. On-line systems are usually simple to use; they afford a personal involvement including correction capabilities upon query results, and they are rapid in response. The use of on-line bibliographic services has accelerated in the seventies largely because of developments in the data communication area which have brought the cost of communication down and increased the geographic availabilities. For example, use of the TYMSHARE network permits users in about 50 major cities to place a local call through the network at a cost of only $10 per hour versus $30 per hour for direct distance dialing (INF,74). The use of communications satellites in the public packet switching networks afford even lower costs, higher speed data transfer, and greater geographic dispersion of information.

One of the most used on-line retrieval system is the MEDLINE system of the National Library of Medicine (NLM). MEDLINE is used by over 120 institutions utilizing the TYMSHARE network for their close to a quarter of a million searches per year.

Data networks have allowed the growth of national information services and, lately, an increasing internationalization of such networks is in evidence. The first official network node outside North America was activated in London in January 1977 as the result of an agreement between Western Union International and the British Post Office. Since

that time, nine more countries have added access nodes to the networks of the United States, and one can begin to see the glimmer of a worldwide information system.

3.6 SUMMARY

Chapter III presented a detailed picture of Information Storage and Retrieval. File structures and access strategies were described. A model of a generalized Information Storage and Retrieval system was presented, and the functional characteristics of its component parts were described. We found that there is a natural relation and interdependence between these parts. The user inputs his query, the Logical Processor transforms it to an internally recognizable form and it performs query expansion, while the Selector retrieves, from the Descriptor file, the set of all documents that are associated with each descriptor in the expanded query. Since the primary concern of an on-line ISR system is response time, the contents of the Descriptor and Document files are organized, using the most suitable file organization for the particular system, to enhance the response. The Locator retrieves from the Document file the entry associated with each documents of the set and returns this information to the user. The Analysis block accepts the new, raw data from the Data block, extracts the pre-defined information and creates two outputs for the Descriptor and Document files. The effectiveness of the whole ISR system is determined by the user when he evaluates the

relevance of the answer to the original question.

Certain individual ISR systems are interconnected and form an information network to enable a user to access any system on the net, be it local or remote.

CHAPTER IV

QUERY ENHANCEMENT

4.1 INTRODUCTION

Precision is determined by the specificity of the keywording, whereas recall is determined by the exhaustivity or depth of indexing. If a person preparing questions for an information retrieval system keeps the above criteria in mind, he can formulate his questions to obtain either high recall or high precision. In addition, by employing vocabulary control, the user will be more assured of having some sort of control over retrieval effectiveness as measured by recall and precision. The understanding of recall and precision should bring the reader to conclude that retrieval effectiveness could be increased by using a thesaurus for vocabulary control. In a computerized information storage and retrieval system, this effectiveness could be still further enhanced by allowing the user to manipulate a thesaurus on-line through a terminal. The user sitting at a terminal, adding to, modifying, displaying a thesaurus or parts of it, in a man-machine interactive environment, would make use of the logic and storage capabilities of the computer and the intelligence of the human. He would have the advantage of being able to extend the thesaurus as he uses it.

Another way of enhancing retrieval effectiveness is when

the user's query is enhanced or expanded for that particular session only. This way his "viewpoint" influences his own query only at that particular time. This gives him the freedom to change, manipulate his own query as he feels it fits his purpose without imposing any bias on the common thesaurus utilized by the rest of the user community. This query expansion methodology is the main aspect of this thesis. Query expansion has its origin in clustering, which was described in general terms in paragraph 3.1.3.1 and will be expanded further in the following sections.

4.2 DOCUMENT VECTOR SPACE

Let us consider a collection of documents D which are represented by terms T, such that

$$Di = (Ti1, Ti2, \ldots . Tin)$$

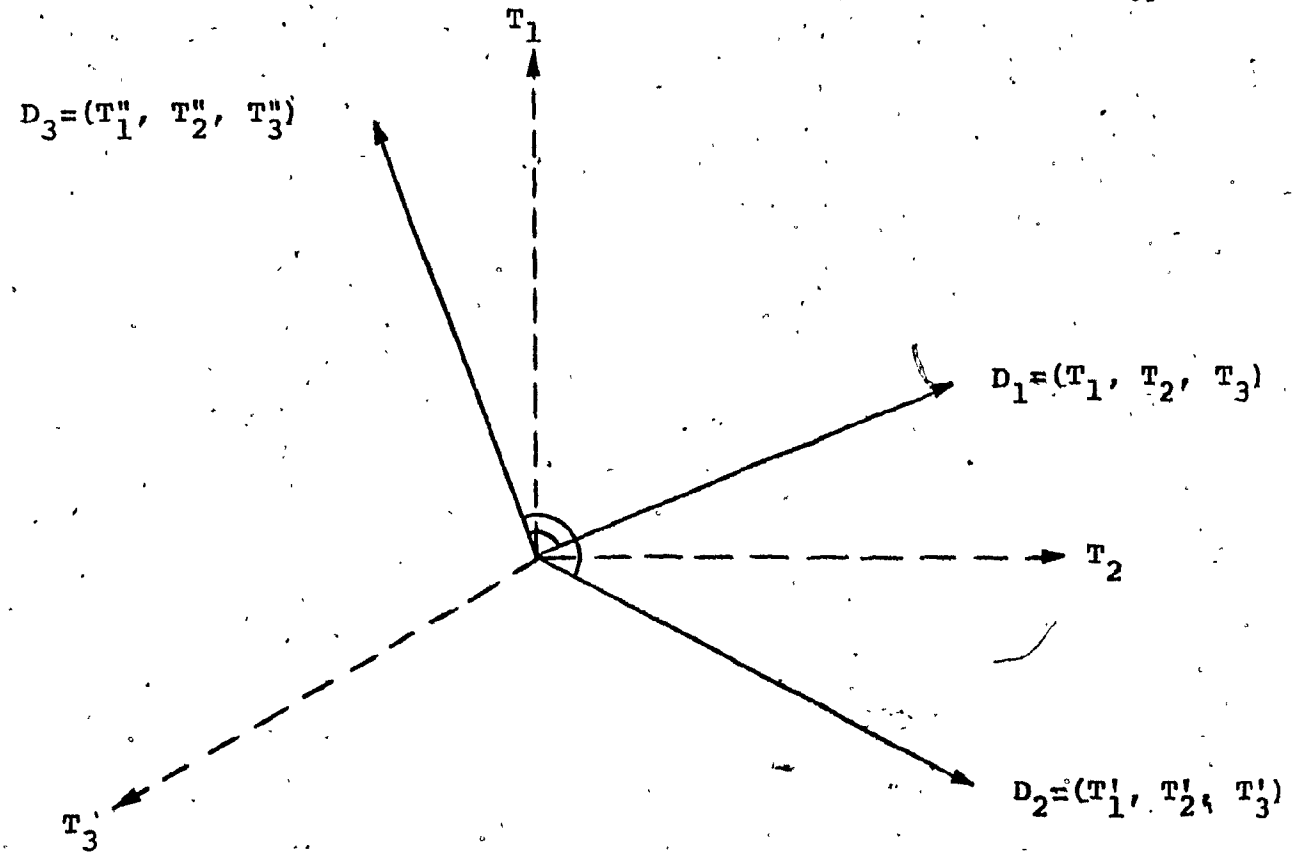
where Tij represents the term j in the vector corresponding to the ith document. It is possible to define a measure of relatedness s(Di, Dj) between documents Di and Dj depending on the similarity of their respective term vectors. In three dimensions, when three terms identify the documents, the situation may be represented by the configuration in figure 4.1. One method to assume similarity between any two of the document vectors would be a function inversely related to the angle between them. That is, when two document vectors are exactly the same, the corresponding vectors are superimposed and the

angle between them is zero.

$$D1 = D2 \text{ and } \Theta = 0$$

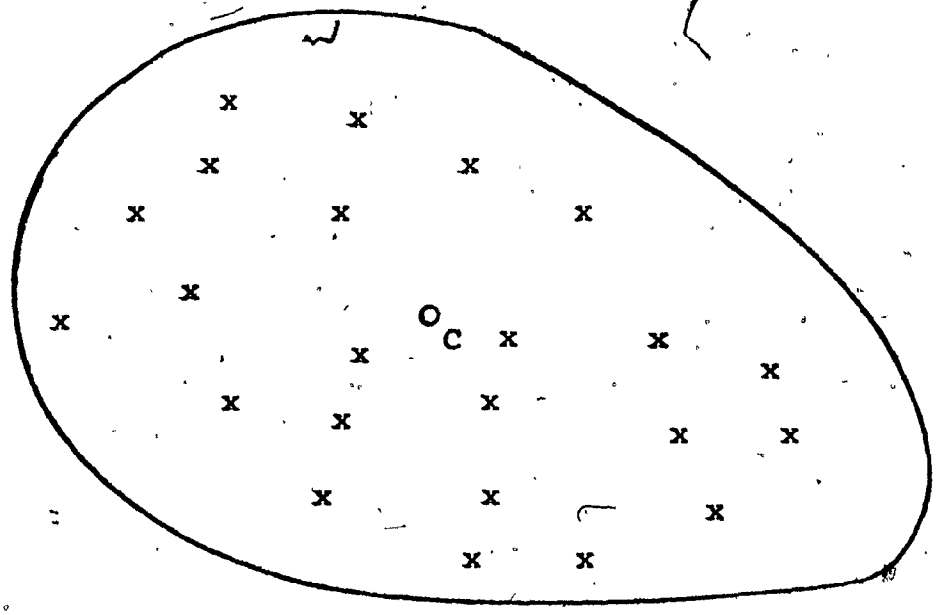
When more than three terms are used to identify a given document, the envelop of the vector space may be used to represent the collection, as shown in figure 4.2. Here only the tips of the document vectors are shown represented by X's, the size of the vectors are normalized to some common length, and the distance between two X's is inversely related to the similarity between the corresponding document vectors; the smaller the distance between X's, the smaller will be the angle between the vectors, and thus the more similar the term assignment.

If a document is introduced into the center of this document space, this central document, or centroid C, can be used to represent the whole collection. Examining the document space configuration of figure 4.2, it shows the details of the indexing choosen for the identification of the documents. This raises the question about the choice of an optimum indexing process, or alternatively, about an effective document space configuration. Studies by G. Salton and A. Wong (SAL,73B), (WON,73) indicate that a good document space is one which maximizes the average separation between pairs of dissimilar documents. In other words, the document space will be maximally separated when the average angular distance between each



VECTOR REPRESENTATION OF DOCUMENT SPACE

FIGURE 4.1



MULTIDIMENSIONAL DOCUMENT SPACE

FIGURE 4.2

document and the space centroid is maximized, that is, when

$$Q = \sum_{i=1}^{n} s(C, Di)$$

is minimum. This similarity measure "s" between two vectors (C, Di) may be assumed to be an inverse function of the angle between them. A typical measure of this type is the cosine function.

When the similarity between two vectors are minimized, it would insure a high precision output, since the retrieval of a given relevant item will then not also entail the retrieval of many non-relevant items in its vicinity.

Salton (SAL,77) describes a particular indexing system known as the discrimination value model, where a good term is assumed to be one which, when assigned as an index term to a collection of documents, will render the documents as dissimilar as possible; that is, it will cause the greatest possible separation between the documents in the document space, decreasing space density. The poor discriminators generally are the ones which will cause the documents to become more similar to each other, and the space density to increase. By computing the space densities both before and after assignment of each term, it was possible to rank the terms in decreasing order of their discrimination values. The discrimination value (DVk) of term k was defined as:

$$DVk = Qk - Q'$$

where Qk is the document density Q with the kth term removed from all document vectors. Obviously, for good discriminators Qk>Q and DVk is positive. The reverse is true for poor discriminators, whose removal causes a decrease in space density, leading to a negative discrimination value. A large majority of the terms are expected to produce neither increase nor decrease in space density, in such a case a discrimination value near zero is obtained.

Using the MEDLAR collection of 450 documents comprising 4726 terms, Salton obtained good retrieval results for his discrimination value model (SAL,75). His graph of terms by document frequency order is reproduced as figure 4.3. Document frequency is defined as the sum of the documents in which a term is present (see Section 4.5). For each class of terms - those of document frequency 1, document frequency 2, etc., - the average rank of the corresponding terms is given in discrimination value order, thereby relating the document frequencies of terms and the corresponding discrimination values. For a set of "t" terms, the discrimination value rank ranges from 1 for the best discriminator to "t" for the worst (i.e., rank 4726 for the Medlar collection).

The U shaped curve of figure 4.3 shows that terms of low document frequency (i.e., occuring only in one, two or three documents) have poor average discrimination ranks. For example, the several thousand terms with document frequency 1, have an

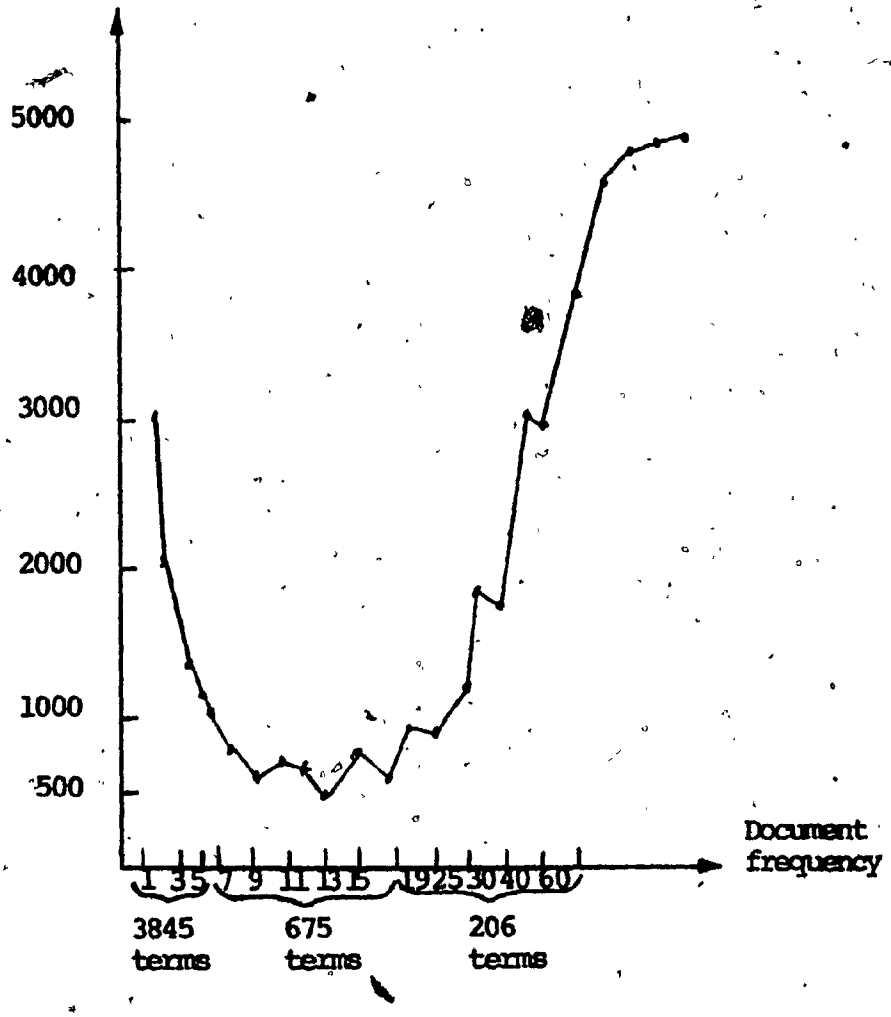
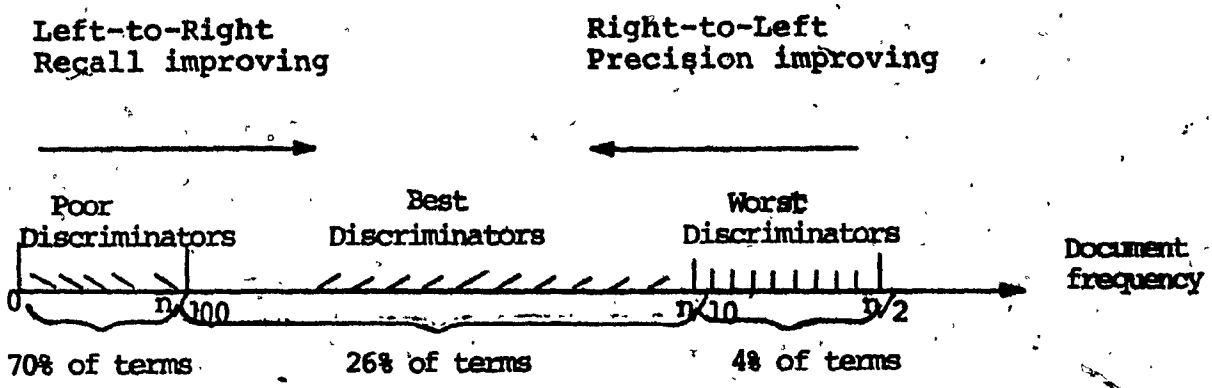


FIGURE 4.3



SUMMARY OF DISCRIMINATION VALUES OF TERMS

FIGURE 4.4

average rank exceeding 3000 out of 4726 in discrimination value order. The terms with document frequency greater than 25 have average discrimination values in excess of 4000. The best discriminators are those whose document frequency is neither too low nor too high. The situation relating document frequency to term discrimination value is summarized in figure 4.4 (after Salton) with the following interpretation.

1. The terms with very low document frequencies, located on the left hand side of figure 4.4, are poor discriminators, with average discrimination value ranks in excess of t/2 for t terms. These terms are so rare and specific that they cannot retrieve an acceptable proportion of the documents relevant to a given query. Their use depresses the recall performance. These terms should be transformed into higher frequency terms, left-to-right on the graph, thus enhancing the recall performance.

2. The terms with high document frequencies exceeding n/10 for n documents, located on the right hand side of figure 4.4, are the worst discriminators, with average discrimination value ranks near t. They are too general in nature, or too broad to permit proper discrimination among the documents. Their use leads to the retrieval of too many items that are extraneous. These terms should be transformed into lower frequency terms, right-to-left on the graph,

thus enhancing the precision performance.

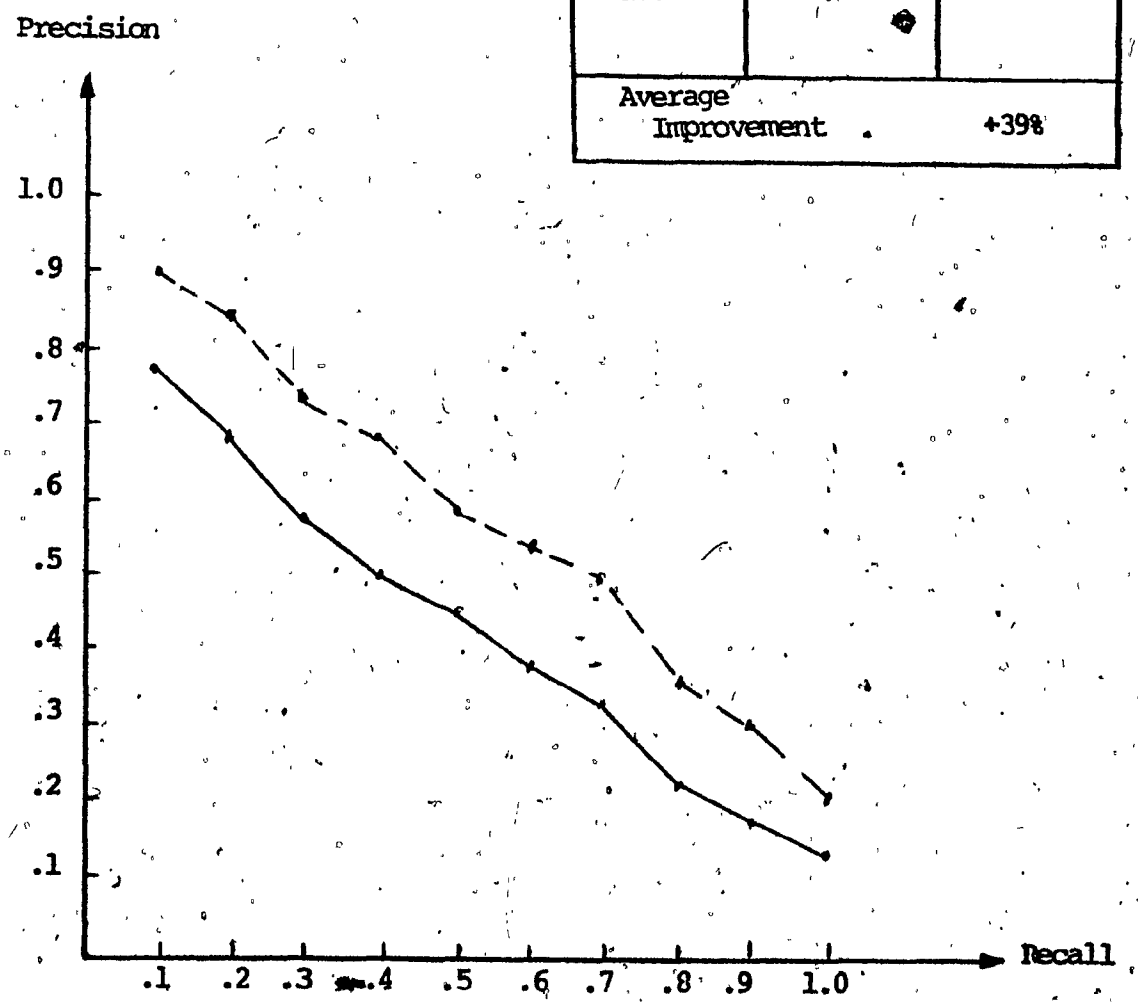
3. The best discriminators are those whose document frequency is neither too high nor too low. With document frequencies between n/100 and n/10 for n documents, the average discrimination value ranks are generally below t/5.

From all this, a useful indexing strategy emerges. Using the right-to-left transformation, the high frequency terms are combined into lower frequency indexing phrases in order to enhance the precision performance of the retrieval system. For example, a phrase such as "programming language" has a lower assignment frequency than either of the high frequency components "language" or "programming".

Using the left-to-right transformation which can be used to enhance recall, low frequency terms with similar properties can be combined into term classes, normally specified by a thesaurus, or synonym dictionary. When a single term is replaced for indexing purposes by a thesaurus class consisting of several terms, the assignment frequency of the thesaurus class will in general exceed any of the components included in the class.

The performance of the right-to-left (phrase) transformation and the left-to-right (thesaurus) transformation for the Medlar medical collection was summarized by Salton (fig. 4.5) and had a 39 percent improvement over the standard term

	Standard Term Frequency	Phrase Assignment
R	Precision	
0.1	.7891	.8811
0.2	.6750	.8149
0.3	.5481	.6992
0.4	.4807	.6481
0.5	.4384	.5930
0.6	.3721	.5450
0.7	.3357	.4867
0.8	.2195	.3263
0.9	.1768	.2767
1.0	.1230	.1969
Average Improvement		+39%



AVERAGE RECALL - PRECISION COMPARISON FOR PHRASES

FIGURE 4.5

frequency process using the unmodified terms. Salton concludes that the space density analysis and the resulting document frequency indexing model appear to perform well for collections in several different subject areas.

4.3 QUERY CLUSTERING

In the previous section, we discussed indexing methodologies which were used to decrease the density of the vectors in the document space. Good retrieval performance was the result of decreased similarity between documents, whereas poor retrieval performance corresponded to document space that was more compressed. It appears that retrieval performance and document space density are inversely related.

On the other hand, the relation between document space and retrieval performance can be considered from the opposite viewpoint. Instead of using indexing methodologies and testing their effect on the density of the document space, the document space configuration can artificially be changed with the submission of queries. Using this approach to improve performance, it would be sufficient to increase the similarity between document vectors located in the same cluster, while decreasing the similarity between different clusters of cluster centroids. The first effect is achieved by emphasizing the terms that are unique to only a few clusters, the second by deemphasizing terms that occur in many different clusters.

Minker et al., (MIN,72) observed that the terms used to index documents tend to occur in clusters, i. e., if term X is used to index a document, then terms Y and Z are likely to be used. Thus, if a query is made to the document space using term X, their system would expand the query to include terms Y and Z also. Based on similarity matrices, clusters were developed and each query was expanded by its cluster related terms. A term was defined to be cluster related if at least one term in the cluster appeared in the original, unexpanded query. That is, if

$$Q = (t1, t2, \ldots . tm)$$

the original query and the cluster developed

$$C = (t2, t4, \ldots . tn)$$

was expanded such that

$$Q \cup C$$

because at least one term (t2) appeared in both, the expanded query became

$$Q' = (t1, t2, t3, \ldots tn)$$

The results were negative; for the specific clustering and matching approach used in their study, it was found that query expansion often had a degrading effect on retrieval performance. Salton noted however (SAL,72C), that in their process the

authors expanded the query, leaving the original terms unchanged, with the main effect of lengthening the query vectors. With longer query vectors, the characteristics of the cosine matching function was altered and therefore careful treatment of adding in new terms to the query is required. In this thesis, the addition of new terms are considered at length.

Sparck - Jones (SPA,71) concluded that a classification system using keywords can be effectively used when only the strong similarity connections are utilized and term groups are used to provide additional rather than alternative description items.

Dattola (DAT,68A) developed a classification scheme which used the document - term matrix directly without the use of a similarity matrix. The experiments showed that documents with many concepts tended to dominate clusters while documents with few concepts were not too dense. He concluded that initial clusters cannot be created in an arbitrary fashion, but must be produced based on some measure of similarity over a part of the document collection.

4.4 PROPOSED APPROACH

The related work in cluster analysis, which has been briefly described in section 4.3 is the basis of the work presented here. We begin with the observation that a reader's apprehension of a document's subject matter is defined by the words making up that document. In this thesis, the process of subject recognition is based on the effectiveness of words representing that document elsewhere than in the document itself. We are investigating the effectiveness of words in the title, abstract, and in the keyword section of documents for the purpose of effective query enhancement. High frequency words are to be extracted from documents, which are classified by a classification system.

These high frequency words are then associated with the classification categories of the system. Each of these categories would constitute a cluster of terms, relevant to that category. It was presumed that certain terms would appear in more than one cluster. This cluster overlap could be utilized in an interactive system, where the user could define the meaning of a term relevant to his query. After defining a single or multiple meaning (category), the contents of the cluster will be printed for the user's query enhancement. The user will choose from these related terms and include them in his modified query, which should return a higher yield of recall and precision.

If the original query Q yields a recall R such that

$$Q = (t1,t2) \Longrightarrow R$$

a modified query Q' will yield a recall

$$Q' = (t1n,t1m,\ldots t2p,t2r,\ldots) \Longrightarrow R'$$

such that: $R<R'$

To facilitate the investigation, a collection of 485 documents from the Communications of the ACM (Association for Computing Machinery), spanning five years from 1967 to 1972, was used. Each document contained the title of the article, the author(s) name, a brief description or abstract, keywords or key phrases used in the article, and classification categories.

The classifications were based on the Computing Review (CR) categories for Computing Sciences. For details, see table 4.1. The general categories are represented by the decimal numbers such as 1., 2., etc., with the name of that category. The fractions such as .2, .22, etc., at the right of the decimal point represented the narrower categories within a main category such as Chemical subcategory (.22) in the general category of Engineering (2.). As it can be observed, there are three levels of hierarchies in the CR category scheme. The structure of this organization, which readily lends itself to our experiment, follows closely the thesaurus relationships of narrower, broader terms as they were explained in section 1.3. Each document

TABLE 4.1

(Reproduction)

CATEGORIES OF THE COMPUTING SCIENCES

Revised Classification System for Computing Reviews

1. GENERAL TOPICS AND EDUCATION
   - 1.0 GENERAL
   - 1.1 TEXTS, HANDBOOKS
   - 1.2 HISTORY, BIOGRAPHIES
   - 1.3 INTRODUCTORY AND SURVEY ARTICLES
   - 1.4 GLOSSARIES
   - 1.5 EDUCATION
     - 1.50 General
     - 1.51 High School Courses and Programs
     - 1.52 University Courses and Programs
     - 1.53 Certification, Degrees; Diplomas
     - 1.59 Miscellaneous
   - 1.9 MISCELLANEOUS

2. COMPUTING MILIEU
   - 2.0 GENERAL
   - 2.1 PHILOSOPHICAL AND SOCIAL IMPLICATIONS
     - 2.10 General
     - 2.11 Economic and Sociological Effects
     - 2.12 The Public and Computers
     - 2.19 Miscellaneous
   - 2.2 PROFESSIONAL ASPECTS
   - 2.3 LEGISLATION, REGULATIONS
   - 2.4 ADMINISTRATION OF COMPUTING CENTERS
     - 2.40 General
     - 2.41 Administrative Policies
     - 2.42 Personnel Training
     - 2.43 Operating Procedures
     - 2.44 Equipment Evaluation
     - 2.45 Surveys of Computing Centers
     - 2.49 Miscellaneous
   - 2.9 MISCELLANEOUS

3. APPLICATIONS
   - 3.1 NATURAL SCIENCES
     - 3.10 General
     - 3.11 Astronomy; Space
     - 3.12 Biology
     - 3.13 Chemistry
     - 3.14 Earth Sciences
     - 3.15 Mathematics; Number Theory
     - 3.16 Meteorology
     - 3.17 Physics, Nuclear Sciences
     - 3.19 Miscellaneous
   - 3.2 ENGINEERING
     - 3.20 General
     - 3.21 Aeronautical, Space
     - 3.22 Chemical
     - 3.23 Civil
     - 3.24 Electrical, Electronic
     - 3.25 Engineering Science
     - 3.26 Mechanical
     - 3.29 Miscellaneous
   - 3.3 SOCIAL AND BEHAVIORAL SCIENCES
     - 3.30 General
     - 3.31 Economics
     - 3.32 Education, Welfare
     - 3.33 Law
     - 3.34 Medicine, Health
     - 3.35 Political Science
     - 3.36 Psychology, Anthropology
     - 3.37 Sociology
     - 3.39 Miscellaneous
   - 3.4 HUMANITIES
     - 3.40 General
     - 3.41 Art
     - 3.42 Language Translation and Linguistics
     - 3.43 Literature
     - 3.44 Music
     - 3.49 Miscellaneous
   - 3.5 MANAGEMENT DATA PROCESSING
     - 3.50 General
     - 3.51 Education, Research
     - 3.52 Financial
     - 3.53 Government
     - 3.54 Manufacturing, Distribution
     - 3.55 Marketing, Merchandising
     - 3.56 Military
     - 3.57 Transportation; Communication
     - 3.59 Miscellaneous
   - 3.6 ARTIFICIAL INTELLIGENCE
     - 3.60 General
     - 3.61 Induction and Hypothesis-Formation
     - 3.62 Learning and Adaptive Systems
     - 3.63 Pattern Recognition
     - 3.64 Problem-Solving
     - 3.65 Simulation of Natural Systems
     - 3.66 Theory of Heuristic Methods
     - 3.69 Miscellaneous
   - 3.7 INFORMATION RETRIEVAL
     - 3.70 General
     - 3.71 Content Analysis
     - 3.72 Evaluation of Systems
     - 3.73 File Maintenance
     - 3.74 Searching
     - 3.75 Vocabulary
     - 3.79 Miscellaneous
   - 3.8 REAL-TIME SYSTEMS
     - 3.80 General
     - 3.81 Communications
     - 3.82 Industrial Process Control
     - 3.83 Telemetry, Missiles, Space
     - 3.89 Miscellaneous
   - 3.9 MISCELLANEOUS

4. SOFTWARE
   - 4.0 GENERAL
   - 4.1 PROCESSORS
     - 4.10 General
     - 4.11 Assemblers
     - 4.12 Compilers and Generators
     - 4.13 Interpreters
     - 4.19 Miscellaneous
   - 4.2 PROGRAMMING LANGUAGES
     - 4.20 General
     - 4.21 Machine-Oriented Languages
     - 4.22 Procedure- and Problem-Oriented Languages
     - 4.29 Miscellaneous
   - 4.3 SUPERVISORY SYSTEMS
     - 4.30 General
     - 4.31 Basic Monitors
     - 4.32 Multiprogramming; Multiprocessing
     - 4.33 Data Base
     - 4.34 Data Structures
     - 4.35 Operating Systems
     - 4.39 Miscellaneous
   - 4.4 UTILITY PROGRAMS
     - 4.40 General
     - 4.41 Input/Output
     - 4.42 Debugging
     - 4.43 Program Maintenance
     - 4.49 Miscellaneous
   - 4.5 PATENTS, SOFTWARE
   - 4.6 SOFTWARE EVALUATION, TESTS, AND MEASUREMENTS
   - 4.9 MISCELLANEOUS

5. MATHEMATICS OF COMPUTATION
   - 5.0 GENERAL
   - 5.1 NUMERICAL ANALYSIS
     - 5.10 General
     - 5.11 Error Analysis, Computer Arithmetic
     - 5.12 Function Evaluation
     - 5.13 Interpolation, Functional Approximation
     - 5.14 Linear Algebra
     - 5.15 Nonlinear and Functional Equations
     - 5.16 Numerical Integration and Differentiation
     - 5.17 Ordinary and Partial Differential Equations
     - 5.18 Integral Equations
     - 5.19 Miscellaneous
   - 5.2 METATHEORY
     - 5.20 General
     - 5.21 Logic; Formal Systems (includes, Boole theorem proving, excludes, switching, lamb
     - 5.22 Automata; finite-state, cellular, stochastic; se machines
     - 5.23 Formal Languages; nondeterministic proces grammars, parsing and translation, abstract l of languages
     - 5.24 Analysis of Programs; schemata; semantics
     - 5.25 Computational Complexity; machine-based; independent, efficiency of algorithms
     - 5.26 Turing Machines, Abstract Processors
     - 5.27 Computability Theory; unsolvability, recursiv
     - 5.29 Miscellaneous
   - 5.3 COMBINATORIAL AND DISCRETE MATHEMATI
     - 5.30 General
     - 5.31 Sorting
     - 5.32 Graph Theory
     - 5.39 Miscellaneous
   - 5.4 MATHEMATICAL PROGRAMMING
     - 5.40 General
     - 5.41 Linear and Nonlinear Programming
     - 5.42 Dynamic Programming
     - 5.49 Miscellaneous
   - 5.5 MATHEMATICAL STATISTICS, PROBABILITY
   - 5.6 INFORMATION THEORY
   - 5.7 SYMBOLIC ALGEBRAIC COMPUTATION
   - 5.9 MISCELLANEOUS

6. HARDWARE
   - 6.0 GENERAL
   - 6.1 LOGICAL DESIGN, SWITCHING THEORY
   - 6.2 COMPUTER SYSTEMS
     - 6.20 General
     - 6.21 General-Purpose Computers
     - 6.22 Special-Purpose Computers
     - 6.29 Miscellaneous
   - 6.3 COMPONENTS AND CIRCUITS
     - 6.30 General
     - 6.31 Circuit Elements
     - 6.32 Arithmetic Units
     - 6.33 Control Units
     - 6.34 Storage Units
     - 6.35 Input/Output Equipment
     - 6.36 Auxiliary Equipment
     - 6.39 Miscellaneous
   - 6.4 PATENTS, HARDWARE
   - 6.9 MISCELLANEOUS

7. ANALOG COMPUTERS
   - 7.0 GENERAL
   - 7.1 APPLICATIONS
   - 7.2 DESIGN, CONSTRUCTION
   - 7.3 HYBRID SYSTEMS
   - 7.4 PROGRAMMING, TECHNIQUES
   - 7.9 MISCELLANEOUS

8. FUNCTIONS
   - 8.0 GENERAL
   - 8.1 SIMULATION AND MODELING
   - 8.2 GRAPHICS
   - 8.3 OPERATIONS RESEARCH, DECISION TABLES
   - 8.9 MISCELLANEOUS

could belong to a number of different categories depending on the classifications of the article at the time of publication.

It was felt that words in the title reflect the contents of the article. The title is usually the author's creation, and as such, it is subjective. Titles may also be misleading, for example W. Churchill's book the "Gathering storm" could be mistaken for a Meteorological text book.

The abstract should also be representative of the article. If automatic abstracting is used, the words with the highest frequencies are presumed to be representing the subject matter, and as such they are utilized in the abstracting process.

The keywords are extracted from the article by a human operator based on a subjective judgement as to what should constitute important keywords.

Since these word sections are representative of the documents' subject matter to a certain degree, it was decided that all three word sections be extracted from each document and, through experiments, the significance of each word group and the interrelations (if any) between them be determined. The words which belong to a particular CR category have some specific significance for that category, in fact they should represent that category or cluster. But how representative these words were, was yet to be determined also. It was decided therefore, that the set of CR categories, which were assigned to

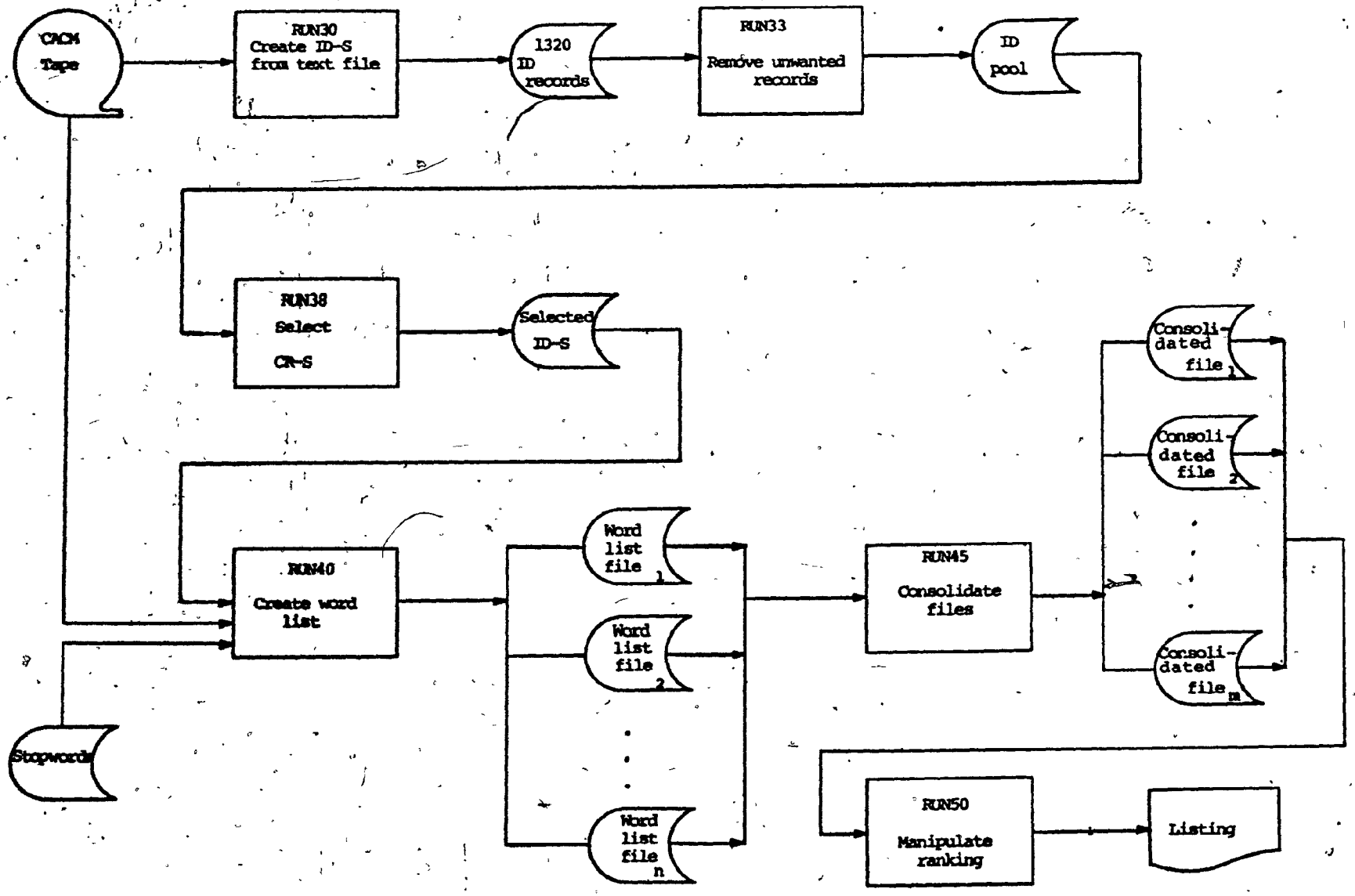
each document will also be extracted. The collected word clusters will then be compared term by term to the corresponding categories of the known NCC (National Computing Center) thesaurus. The comparison between the two should indicate:

1. how representative the collected clusters from the CACM are,
2. whether these clusters are more/less representative than the base they are compared to, and
3. whether these clusters would represent an enhancement to the original query, if they were selectively inserted into the query.

4.5 METHODOLOGY

A set of programs was developed (see Appendix A) on the Concordia University's CDC 6400 Time-sharing system to facilitate the study. The procedure consisted of the following steps (see figure 4.6).

1. The 485 documents of the CACM were processed (Run30), rejecting 23 of them due to errors which could not be resolved. For each CR category in a document, an identity (ID) record was built using the CR category, the year of publication, the volume and page numbers. If, for example, a document contained five CR categories, five ID records were built. The 462



CACM DATA EXTRACTION SYSTEM DIAGRAM

FIGURE 4.6

documents reaching this stage of the processing yielded 1322 records in 171 different categories. After removing records with erroneous and nonexistent categories, the file was sorted (Run33) and a pool of ID records was created for all the CR categories in use.

2. To select one CR category on a narrow term level (such as 4.21 - Machine oriented languages), or any number of CR categories on an intermediate or broader term level (such as 4.00 - Software), the categories were selected from the ID pool (Run38). These selected ID records - containing the publication year, volume and page numbers of a document - were matched against the CACM document tape (Run40). Documents matching the ID records were processed, and the words were extracted from the title, abstract and keyword section of the document. Before a word list was created, each word was matched against a "stopword" file containing 1848 noise words (see table 4.2 for a sample of noise words). The word list created this way contained all the words in that particular CR category detailing the number of times each appeared in the title, abstract and keyword sections of the document. In addition to these, the following two parameters were also computed and printed (see Appendix C for the five parameters).

2.1 Total frequency, F(tot), which is defined as the

TABLE 4.2

LISTING FROM THE STOP WORD FILE

GREAT	OLD	YEAR	OFF	COME	SINCE	AGAINST
CAME	RIGHT	USED	TAKE	THREE	STATES	HIMSELF
HOUSE	USE	DURING	WITHOUT	AGAIN	PLACE	AMERICAN
HOWEVER	HOME	SMALL	FOUND	MRS	THOUGHT	WENT
PART	ONCE	GENERAL	HIGH	1	UPON	SCHOOL
DOES	GOT	UNITED	LEFT	NUMBER	COURSE	WAR
ALWAYS	AWAY	SOMETHING	2	FACT	THOUGH	WATER
PUBLIC	PUT	THINK	ALMOST	HAND	ENOUGH	FAR
HEAD	YET	SYSTEM	BETTER	SET	TOLD	NOTHING
END	WHY	CALLED	EYES	FIND	GOING	LOOK
LATER	KNEW	POINT	NEXT	PROGRAM	CITY	BUSINESS
GROUP	PRESIDENT	TOWARD	YOUNG	DAYS	LET	ROOM
SOCIAL	GIVEN	PRESENT	SEVERAL	ORDER	NATIONAL	DINNER
SECOND	FACE	PER	AMONG	FORM	OFTEN	THINGS
EARLY	WHITE	CASE	JOHN	BECOME	LARGE	BIG
FOUR	WITHIN	FELT	ALONG	CHILDREN	SAW	BEST
EVER	LEAST	POWER	LIGHT	THING	SEEMED	FAMILY
WANT	MEMBERS	MIND	COUNTRY	AREA	OTHERS	DONE
ALTHOUGH	OPEN	GOD	SERVICE	CERTAIN	KIND	PROBLEM
DIFFERENT	DOOR	THUS	HELP	SENSE	MEANS	WHOLE
PERHAPS	ITSELF	YORK	TIMES	HUMAN	LAW	LINE
NAME	EXAMPLE	ACTION	COMPANY	HANDS	LOCAL	SHOW
HISTORY	WHETHER	GAVE	EITHER	TODAY	ACT	FEET
3	PAST	QUITE	TAKEN	ANYTHING	HAVING	SEEN
BODY	HALF	REALLY	WEEK	CAR	FIELD	WORD
ALREADY	TELL	COLLEGE	SHALL	TOGETHER	MONEY	PERIOD
KEEP	SURE	PROBABLY	FREE	REAL	SEEMS	BEHIND
MISS	POLITICAL	AIR	QUESTION	MAKING	OFFICE	BROUGHT
SPECIAL	HEARD	MAJOR	PROBLEMS	AGO	BECAME	FEDERAL
STUDY	AVAILABLE	KNOW	RESULT	STREET	ECONOMIC	BODY
REASON	CHANGE	SOUTH	BOARD	JOB	SOCIETY	AREAS
CLOSE	TURN	LOVE	COMMUNITY	TRUE	COURT	FORCE
COST	SEEM	AM	WIFE	AGE	FURTHER	VOICE
CENTER	WOMAN	COMMON	CONTROL	NECESSARY	FOLLOWING	POLICY
SOMETIMES	GIRL	SIX	CLEAR	FURTHER	LAND	ABEL
MUSIC	PARTY	PROVIDE	EDUCATION	CHILD	EFFECT	LEVEL
MILITARY	RUN	SHORT	STOOD	TOWN	MORNING	TOTAL
FIGURE	RATE	ART	CENTURY	CLASS	NORTH	USUALLY
PLAN	THEREFORE	EVIDENCE	MILLION	SOUND	TOP	BLACK
STRONG	VARIOUS	BELIEVE	PLAY	SAYS	SURFACE	TYPE
MEAN	SOON	LINES	MODERN	NEAR	PEACE	TABLE
ROAD	TAX	MINUTES	PERSONAL	PROCESS	SITUATION	4
ENGLISH	GONE	IDEA	INCREAS	NOR	SCHOOLS	WOMEN
LIVING	STARTED	BOOK	LONGER	CUT	DR	FINALLY
PRIVATE	SECRETARY	THIRD	MONTHS	SECTION	CALL	ENTIREL
EXPECTED	FIRE	NEEDED	GROUND	KEPT	VALUES	VIEW
PRESSURE	BASIS	SPACE	EAST	FATHER	REQUIRED	SPIRIT
COMPLETE	EXCEPT	MOVED	WROTE	CONDITION	RETURN	SUPPORT
LATE	RECENT	HOPE	LIVE	BROWN	COSTS	ELSE
FORCE	HOURS	NATIONS	PERSON	TAKING	COMING	DEAD
LOW	MATERIAL	REPORT	STAGE	DATA	HEART	INSTEAD
LOST	MILES	READ	ADDED	AMOUNT	FEELING	FOLLOWE
PAY	SINGLE	BASIC	COLD	HUNDRED	INCLUDING	INDUSTR

total number of times a term occurs in all documents of the database.

$$F(tot) = \sum_{i=1}^{n} \sum_{j=1}^{m} tij$$

where n = no. of documents

m = no. of terms in document i

2.2 Document frequency F.(df), is the number of documents in which a term is present, regardless of the number of occurrences.

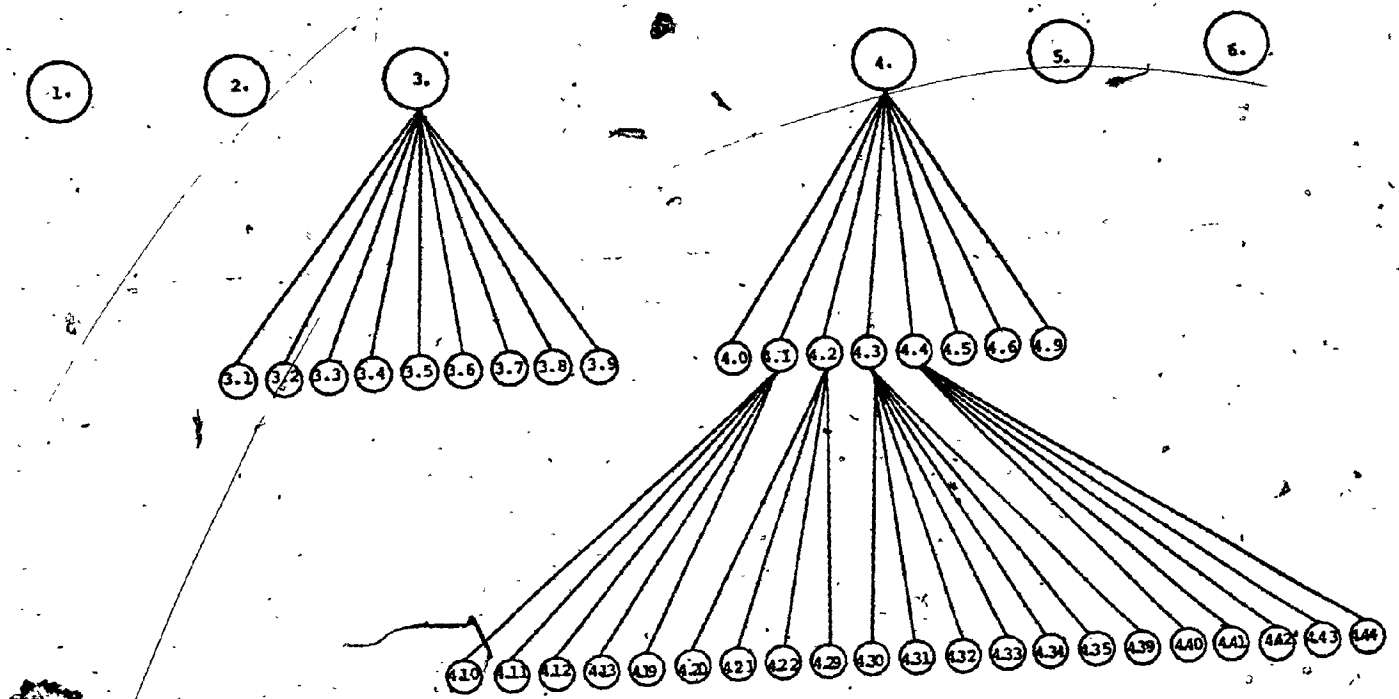
$$F(df) = \sum_{i=1}^{n} ti$$

where n = no. of documents in the data base

$$ti \begin{cases} = 1 \text{ if term is present in the document} \\ = 0 \text{ otherwise} \end{cases}$$

After a number of runs, word lists were created on all levels and categories of the published list of Computing Reviews (see figure 4.7). The number of documents in each main category were the following.

1 -	General Topics	63 doc.	4.7%
2 -	Computing Milieu	48 -"-	3.6%
3 -	Applications	352 -"-	26.7%
4 -	Software	408 -"-	31.0%
5 -	Math. of Computation	292 -"-	22.0%
6 -	Hardware	78 -"-	6.0%
X -	Miscellaneous	79 -"-	6.0%



HIERARCHIES OF TERMS PROCESSED

FIGURE 4.7

3. The low level word list files were consolidated into mid and high level files (Run45). Since each term has five parameters describing its different occurrences in the title, abstract, etc., these parameters were used to manipulate the ranking of terms in different orders (Run50). To determine the significance of word sections individually and their interrelations, each file was sorted eight different ways in an ascending order on the following parameter combinations.

   Title

   Title - Abstract

   Title - Keyword

   Abstract

   Abstract - Keyword

   Database frequency

   Document frequency

   Alpha

4.6 EXPERIMENTAL FINDINGS

This section presents the results obtained, and some preliminary conclusions drawn from the investigation. Since the CR category on Software constitutes the largest portion (31%) of the document selection, it was decided that the detailed investigation proposed for this thesis will be carried out on this category.

1. Total frequency. When low level categories printed in total frequency rank order were examined, the words with the highest ranks were fairly relevant to their respective categories. For example category 4.12 - Compilers and generators, contained the following high ranking terms (4.12, 4.19 are not in the Appendix).

Algorithm	Grammars	Compiler
Matrix	Syntax	Parsing
Processor	Array	

For category 4.19 - Miscellaneous Processors, the following high ranking terms were found.

Tables	Entry	Storage
Automatic	Collector	Segmentation
Garbage	Virtual	

When medium and high level categories were examined,

the high ranking terms were still representative of their categories.

As the categories became more general on the mid and high level, the terms belonging to these categories became more general also. For the high level selection of category 4.00 - Software for example, the following high ranking terms were found (for more details, read Appendix B horizontally).

Computer	Software	Programming
Algorithm	Storage	Information
Sharing	Paging	

From the above it is concluded that whether a term resides in the title, the abstract or the keyword section of a document, the total frequency of the term is fairly representative of its category both on the specific, narrow and general; high level.

2. Document frequency. When a combination of terms represents a subject category in a set of documents, it can be deduced that, in general, the document frequencies are representative of the subject categories of the data base. The more documents a term appears in, the more those documents will represent the category of the term. In other words a high document frequency represents a concentration of

documents in a particular subject. This empirical conclusion, however, could not be substantiated with the present set of data.

3. Title, abstract, keyword sections. Using mid level categories, the different word sections were examined. It was found that high frequency terms in the keyword section were very special terms. Using table 4.3, it can be seen that for category 4.1 - Processors, the words like paging, processing, linkage are special. Looking at category 4.2 - Programming languages, words as simulation, algorithmic, induction are very representative, so are paging, multiprocessor, and segment for category 4.3 - Supervisory systems. As can be seen, this situation occurs when the term frequencies are low in the title and abstract, and high in the keyword section of the word list selection.

At the low frequency end of the keyword section, some special words were found. These words most infrequently seem to appear in the other two sections, as can be seen in table 4.4. For category 4.3 - Supervisory systems, words as interlocks, semaphores or for category 4.2 - Programming languages, words as ring, nanoprogram, partition are very special terms. Combining these words would result in the loss of their special status. It is concluded therefore, that

TABLE 4.3

HIGH FREQUENCIES IN THE KEYWORD SECTION

(F(tit) = low, F(abs) = low, F(key) = high)

Terms	Title	Abstract	Keyword
Paging	2	7	20
Syntax	0	7	13
Processing	1	6	11
Compiler	0	6	10
Simulation	1	4	7
Multiprocess	1	1	6
Manipulating	2	2	6
Models	1	2	6

Note: The high frequencies indicated in these tables are the highest frequencies of a sort on the particular field(s) and as such they should be considered as frequencies with relatively high values or frequencies that are greater than zero.

TABLE 4.4

SPECIAL TERMS OF LOW FREQUENCY

$$(F(tit) \leq F(key) \leq F(tot) \leq 3) \cap (0 = F(abs) < F(tit))$$

Terms	Title	Abstract	Keyword
Metacompiler	0	0	3
IITKAN	1	0	2
Linkage	0	0	3
Bootstrap	0	0	2
Generator	1	1	2
Symbol	1	1	2
Loader	1	0	2
Ring	1	0	1
Recursion	1	0	1
Pseudo	1	0	1

these are very special terms of low frequency, and they cannot be captured without the keyword section. Their frequency range is defined as:

$$\{F(tit) <= F(key) <= F(tot) <= 3\} \wedge \{0 = F(abs) < F(tit)\}$$

where F = frequency
tit = title
key = keywords
tot = total
abs = abstract

Looking at high frequency terms in the abstract section (table 4.5) it can be seen that these words are not special words, but fairly common words or qualifiers. For example, for category 4.1 - Processors, words as languages, user, criteria, or for category 4.2 - Programming languages, words such as implementation, instruction, method are words that are used only to qualify some meaning.

Using table 4.6 to examine high frequency terms in the titles, it is easy to realise that these words are only noise words and could be put in almost any context without loosing their generalities.

4. Interaction between the title, abstract, and keyword sections. When a high frequency combination of any

TABLE 4.5

HIGH FREQUENCIES IN THE ABSTRACT SECTION

(F(tit) = low, F(abs) = high, F(key) = low)

Terms	Title	Abstract	Keyword
Computer	3	32	14
Processes	2	23	7
Grammars	5	22	5
User	1	19	4
Information	2	18	7
Implementation	1	17	1
Programming	1	16	8
Structures	0	14	5
Algorithm	2	14	3
Hardware	0	14	0
Software	1	14	5
Language	1	13	3
Communication	4	12	5
Execution	0	11	0
Definition	1	11	1

TABLE 4.6

HIGH FREQUENCIES IN THE TITLE SECTION

(F(tit) = high, F(abs) = low, F(key) = low)

Terms	Title	Abstract	Keyword
Comment	2	0	0
Estimating	1	0	0
Analysing	1	0	0
Memo	1	0	0
Anomaly	1	0	0
Locality	1	0	0
Compute	1	0	0
Approach	1	0	0

two word sections is examined, the result is not as straightforward as it was above. Using table 4.7, let us start with the case of high frequencies in the keyword and abstract sections. The terms are generally descriptive of their category. In order to see the effect of the varying frequencies, we plot the abstract F(abs) and keyword F(key) frequencies on a graph. Since the two sets of frequencies increase in parallel, they will be distributed along a conical shape in the middle of the graph. Using our previous results of high frequency keywords F(key) of table 4.3, we can superimpose these values on the X axis of the graph. Similarly, with the aid of table 4.5, the F(abs) values can be plotted on the Y axis of the graph. The total interaction between the keyword and abstract sections is summarised in figure 4.8.

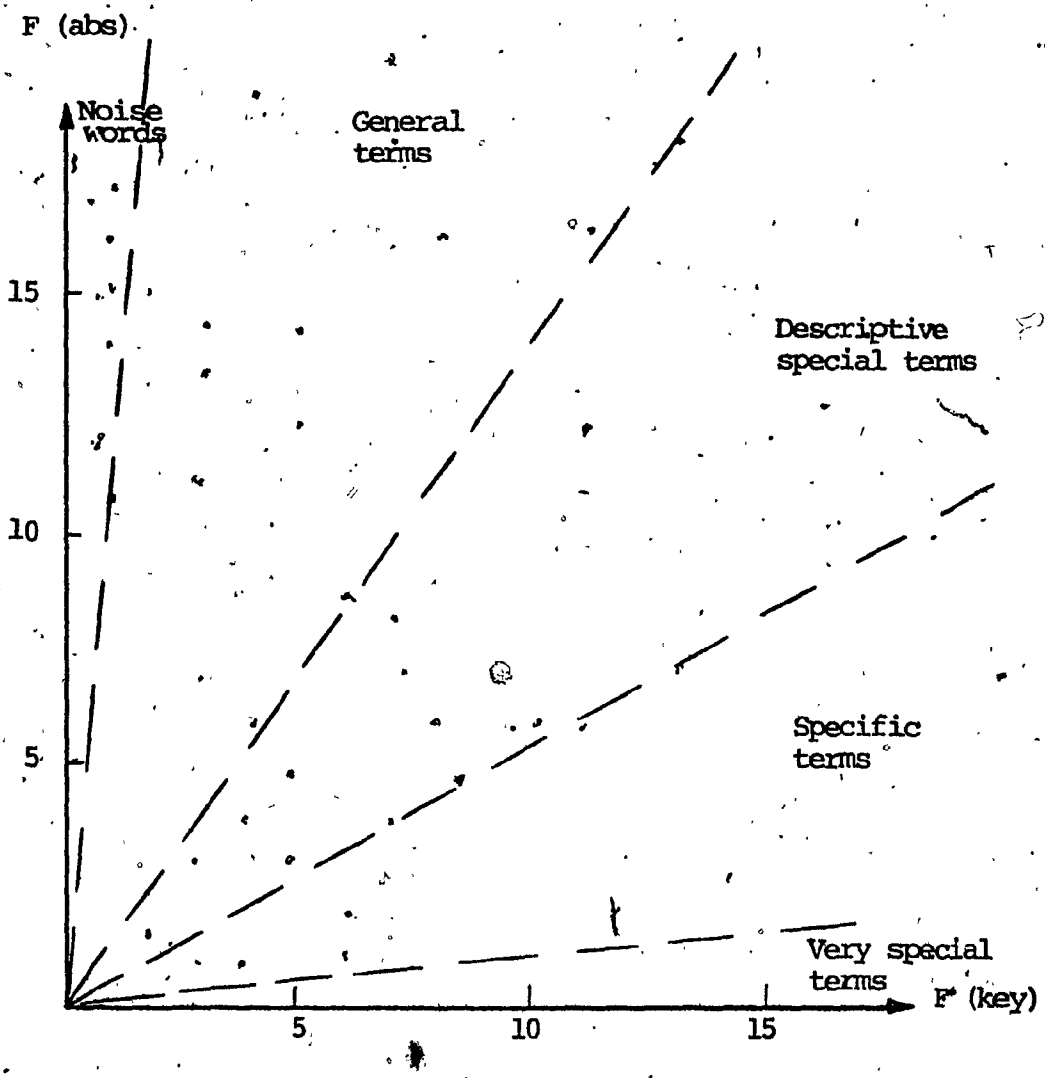
From this figure, we can see that terms with relatively equal frequencies in both the keyword and abstract sections are descriptive, special terms. As the frequency in keywords decreases, there is a band of general terms first, then a narrower band of noise words. In the other way, when the frequency in the abstract decreases, there is a band of special words and beyond that, some low frequency, very special words.

TABLE 4.7

HIGH FREQUENCIES IN THE ABSTRACT AND KEYWORD SECTIONS

(F(tit) = low, F(abs) = high, F(key) = high)

Terms	Title	Abstract	Keyword
Programming	6	80	20
Storage	6	24	21
Sharing	5	20	23
Computer	3	18	13
Languages	4	14	17
Scheduling	1	16	11
Performance	3	12	11
Compiler	0	8	7
Syntax	1	6	10



FREQUENCY INTERACTION: ABSTRACT VS. KEYS

FIGURE 4.8

The high frequencies for the title and abstract combination are not really high, as shown in table 4.8. Using the same procedure as before, a similar graph can be developed to show the interaction between title and abstract frequencies (fig. 4.9). From this graph, we can see that there is a wide band of qualifiers from the abstract section. In the middle of the graph, where most of the intersection should take place, there are only general terms and the interaction is weak judging from the magnitude of the cone. There are some noise words associated with the title axis, but there is a large no-man's land, indicating again little interaction between the two word sections.

High frequencies for the title and keyword combination yielded even lower frequencies than the title and abstract combination pair. Plotting a graph seemed futile, because the magnitude of the interacting frequencies never exceeded 2 (see table 4.9). For this combination pair, it seems that there is very little interaction; this perhaps could be attributed to the fact that terms in the title section are a subset of the terms in the keyword section.

$$T(tit) \in T(key)$$

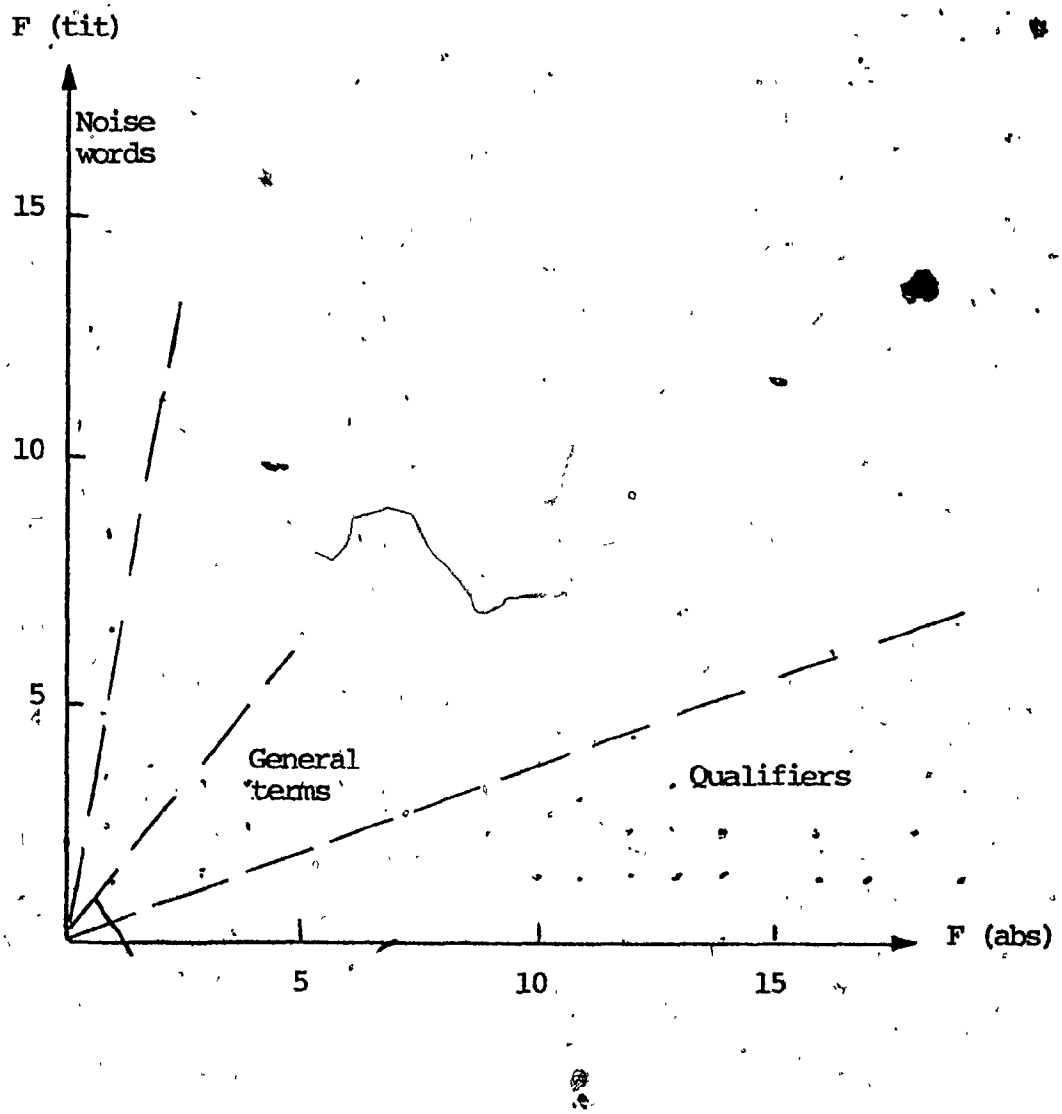
Table 4.10 lists all the word section combinations and

TABLE 4.8

HIGH FREQUENCIES IN THE TITLE AND ABSTRACT SECTIONS

(F(tit) = high, F(abs) = high, F(key) = low)

Terms	Title	Abstract	Keyword
Generating	3	4	0
Interactive	3	3	0
Paged	3	3	1
Environment	3	3	1
Hash	1	1	0
Nucleus	1	1	0
Timing	1	1	0



FREQUENCY INTERACTION: TITLE VS. ABSTRACT

FIGURE 4.9

TABLE 4.9

HIGH FREQUENCIES IN THE TITLE AND KEYWORD SECTIONS

(F(tit) = high, F(abs) = low, F(key) = high)

Terms	Title	Abstract	Keyword
Debugging	1	0	3
Instrument	1	0	2
Multiple	1	0	1
Pseudo	1	0	1
Recursion	1	0	1
Device	1	0	1

TABLE 4.10

SUMMARY OF THE WORD SECTION COMBINATIONS

Conclusions	Title	Abstract	Keyword	Table
Trivial, ignore	low	low	low	-
Very special terms. Cannot be captured without keywords	low	low	high	4.3
Generally descriptive words	low	high	high	4.7
Qualifier, writer style	low	high	low	4.5
Noise Words - inconclusive	high	low	low	4.6
High frequency never occurs - inconclusive	high	high	low	4.8
High frequency never occurs - inconclusive	high	low	high	4.9
Trivial, ignore	high	high	high	-

summarizes the corresponding conclusions made above.

5. Multiple cluster appearance of terms. Terms, which were common to all or most of the CR sub-categories, were found to be high frequency terms. See the high frequency words in Appendix C. Basing our conclusion on the characteristics of high frequency terms, if these high frequency common terms were used in query enhancement, they would yield a low precision (P), but high recall (R) due to the general nature of the terms. If term tk appears in Gm groups, where m is the maximum number of groups, then

$$Q' = (t1, t2, ..tk, ..) \Longrightarrow R'$$

such that,

$$R<R' \text{ and } P>P'$$

6. Comparison of CR categories to the NCC thesaurus. On the mid level, CR category clusters were compared term by term to the corresponding CR categories of the NCC thesaurus. To set up a CR term list from the NCC, the following process was used. Using CR category 4.3 - Supervisory Systems as an example (see table 4.1), each low level category such as 4.31, 4.32, etc., was looked up in the alpha section of the NCC. Under Multiprocessing for instance, the following terms were found.

3538 Multiprocessing systems

NT 7011 Dual computer systems

5772 Multiprocessing software

2626 Multiprocessor architecture

4614 Parallel - processor systems

With the help of the reference number, these terms were traced in the hierarchy list and the associated terms were collected for the term list. Under Real - time computers, for example, the following terms appeared.

528 Real time computers

3693 Real time software

2815 Real time operating system

529 Real time supervisors

Once the NCC term list (base cluster) was assembled, it was compared to the mid level alpha list (test cluster) of the CACM, term by term. Terms from the base cluster with no match in the test cluster were noted and the occurrences were counted. Table 4.11 shows the no match terms for the comparison between

TABLE 4.11

PARTIAL LIST OF NO MATCH TERMS OF CATEGORY 4.3

Time	Business
CPL/1	JOSS
PM3	TELCOMP
BASIC	CALL/360
COBOL	DELTA
LRLTRAN	FOCAL
PL/1	POP-2
Consolidators	Audit
Syntax	Plotting
Directed	Preprocessor
On-line	Decision
Table	Precompiler
Associative	Conduct
Pushdown	Ring
Tree	Management

the base and test clusters of the 4.3 category. These are mostly very unique terms with a low recall, but high precision characteristics.

7. Measure of cluster effectiveness. In order to determine the effectiveness of the test clusters with respect to the base, a comparative measure was developed. Consider cluster A, a test cluster and cluster N, the base cluster. The measure of the cluster effectiveness E is defined as

$$E = \frac{A \wedge N}{N}$$

where: $A \wedge N$ are the common terms or hits

The result of the comparison between test and base clusters, using the measure of cluster effectiveness, are tabulated in table 4.12. In these categories the measure of E as an average is 71%, which makes the test files effective. On the other hand, this high percentage could be attributed to the relatively large test files and not necessarily to the subject matter being common.

8. Average frequency. It was found that the largest portion of the terms common to the test and base clusters were in the top frequencies of the test clusters. Using the CR category 4.2 test cluster as an example, 70% of the common terms were lying within the top 247 frequency ranks, which represented 25% of

TABLE 4.12

CLUSTER EFFECTIVENESS

CR	A	N	A/N	E(%)
4.1	869	361	280	77
4.2	926	342	228	67
4.3	950	231	169	73
4.4	952	86	59	69

all terms.

The average frequency of a term was defined as:

$$F(ave) = \frac{F(tot)}{F(db)}$$

and the average of these F(ave) frequencies over a range "n" was defined as:

$$F = 1/n \sum_{i=1}^{n} Fi(ave)$$

For the 4.2 test cluster, the average frequencies ranged between 1 and 6. Over the range of the 247 top ranking terms, the average of the F(ave) frequency was found to be 2.9. For terms below this ranking, the average of the F(ave) frequency was 1.5.

It can be seen that 70% of the common terms, lying in the top 25% of the test cluster, appeared close to three times in the CACM, whereas the remaining 30% of the common terms, appearing 1.5 times only, were scattered in the low fequency end of the test cluster, occupying 75% of the file. We can conclude therefore that most of the relevant terms for the CACM test file were in the top 25% of the file and, except for the few, very special words in the low frequency range, the terms in the rest of the file (75%) were negligible.

CHAPTER V

CONCLUSIONS

In this chapter, the main conclusions of the research will be given and the overall contribution of the thesis will be considered.

5.1 CONCLUSIONS

The main aim of this research was to investigate the effectiveness of terms in the title, abstract, and the keyword sections of documents and to develop an effective query enhancement methodology for thesaurus manipulation in an on-line environment. The results of this investigation (see Appendix B and C) give some heuristic recommendations for query modification. These recommendations could be utilized in an on-line system to help the user to select the most relevant terms for his query enhancement.

The document structure used in the research was the structure of the Communications of the ACM (CACM); that is, it contained titles, keywords, abstracts and authors.

Terms that are common to a number of clusters are not very effective in query enhancement. If we assume that documents within a cluster are normally showing identical, or close relevance characteristics, then the best retrieval performance

should be obtained with a cluster space exhibiting tight individual clusters, but large intercluster distances. In order to achieve this, we select terms with high discrimination value (DV), as this was explained in section 4.2.

High frequency terms are also considered poor to mediocre in query enhancement. Although they are relevant to their categories, these terms tend to be fairly general. Terms with medium frequencies are good discriminators and using them should yield higher precision. We found some very special terms at the low end of the frequency scale which, if not handled in a special way, would be lost for query enhancement. In general though, most of the low frequency terms are too rare to retrieve an acceptable proportion of the documents relevant to a query. The above conclusions bear out Salton's "Term Weighting" system, which proposes to assign the largest weight to those terms with high frequency in individual documents, but which are at the same time relatively rare in the collection as a whole (SAL,75).

Examining the contents of the abstract section, it seems that these words reflect the writer's style. The fact that they appear in the abstract shows that, to the writer, they are important. The study of high frequency terms in the abstract section reveal, however, that these terms are common, qualifying words, with no special significance attached to them. A balanced interaction was found between the abstract and keyword sections as the function of frequencies.

Terms in the title seem to have a close relationship with terms in the keyword section. Most words in the title are used in the keyword section as part of phrases. We conclude therefore, that terms in the title section are a subset of the terms in the keyword section.

High frequency terms in the keyword section are good, combinable words for phrases. Keywords are very special terms and in a normal query, they could be totally missed if they are not part of the retrieval documents' title. Although keywords are selected by the author or other human operator on a subjective basis as to the importance of the terms, these keywords contribute the most information about a document. A conclusion therefore is evident. If we want to economise and retrieve the most relevant documents, we should use terms from the keyword section for query enhancement.

From keywords alone, however, it is hard to distinguish the useful terms in a writer's style. On the other hand, if we supplement the keywords with the abstracts, we can determine the useful terms from this combination. It is suggested therefore that future data-bases include abstracts also as one of the document surrogates.

The comparison of the CACM test clusters to the NCC base clusters revealed that the test clusters contained, as an average, 71% of the base clusters' terms. This high coincidence is considered representative and it is concluded that the terms

in the test clusters would represent an enhancement to the original query if they were selectively inserted. It should also be noted that 70% of these common terms were in the top 31% of the test clusters, and if one is willing to trade the 30% remaining common terms for economic savings, one can save, as an average, 69% of the processing in any test file.

5.2 CONTRIBUTION OF THIS THESIS

The main contribution of this thesis has been the development of an economical, yet effective query enhancement methodology. Since the basic elements of the proposal are in existence in most on-line information retrieval system, it is felt that the new way of utilizing existing resources would make a viable economic proposition. It is recognized however that, to implement this query enhancement, further detailed work is necessary.

Frequencies in title, abstract and the keyword sections of the document data base were investigated and the interactions between the three sections were defined. The outcome of this investigation was the proposed utilization of terms from the keyword section for query improvement.

To fulfill the requirements set out for the research in Section 4.4, a measure of cluster effectiveness was developed to compare the NCC base and CACM test clusters. This measure was used to prove that the test clusters are representative and that the proposed methodology is effective for query enhancement.

BIBLIOGRAPHY

AIT,69 Aitchison, J., Day, P. "Thesaurofacet: A Thesaurus and Faceted Classification for Engineering and Related Subjects". Whetstone, England, English Electric Company Ltd., 1969.

ANN,78 Annual Review of Information Science and Technology. Published by Knowledge Industry Publication Inc., Vol. 13, page 8

BEN,75 J. L. Bentley, "Multidimensional Binary Search Trees Used for Associative Searching". Communications of the ACM, Sept., 1975.

BON,64 R. E. Bonner, "On Some Clustering Techniques". IBM Journal of Research and Development, 8(No. 1): 22-32, 1964

COS,66 Costello, J. C. Jr., "Systems for the Intellectual Organization of Information, Volume VII, Coordinate Indexing". New Brunswick, New Jersey, The Rutgers University Press, 1966, page 90.

CRO,71 Croghan, A., "A Manual on the Construction of an Indexing Language Using Educational Technology as an Example". London, Coburgh Publications, 1971.

CRU,71 Crouch, C. J., "Language Relations in a Generalized Information Storage and Retrieval System". Southern Methodist University, Ph. D., 1971. University Microfilms 72-16, 303., Ann Arbor, Michigan.

DAT,68 R. T. Dattola, "A Fast Algorithm for Automatic Classification" Report No.: ISR-14, National Science Foundation, Computer Science Dept., Cornell University, 1968.

DAV,68 Davis, C. H., "Integrating Vocabularies with a Classification Scheme", American Documentation Vol. 19, No. 1, page 101, January 1068.

DIM,73 J. J. Dimsdale, H. S. Heaps, "File structure for an On-line Catalog of One Million Titles", Journal of Library Automation, Vol. 6, March 1973.

DOY,65 Doyle, L. B., "Expanding the Editing Function in Language Data Processing". CACM 1965 8(4).

ERI,68 Thesaurus of ERIC Descriptors, Washington, D. C., U. S. Government Printing Office, 1968.

FIN,74 Finkel, R. A., and Bentley, J. L., "Quad Trees: A Data Structure for Retrieval on Composite Key". Acta Informatica 4,1974.

HEA,70 Heaps, H. S. and Thiel, L. H., "Optimum Procedures for Economic Information Retrieval". Information Storage and Retrieval 6:2 (June 1970).

HOY,73 W. G. Hoyle, "Automatic Indexing and Generation of Classification Systems by Algorithm". Information Storage and Retrieval, Vol. 9, pp. 233-242. Pergamon Press, Great Britain, 1973.

HUF,52 D. A. Huffman, "A Method for the Construction of Minimum Redundancy Codes", Proc. I. R. E. 40, Sept., 1952.

INF,74 The Annual Review of Information Science and Technology (Vol. 9, page 287).

KAZ,80 E. J. Kazlauskas and T. D. Holt, "The Application of a Minicomputer to Thesaurus Construction". Journal of the American Society for Information Science, Washington, D. C., 1980.

KNU,73 Donald E. Knuth, "The Art of Computer Programming", Published by Addison - Wesley Publishing co., 1073.

MAR,75 James Martin, "Computer Data Base Organization", Published by Prentice Hall Inc., Englewood Cliffs, N. J., 1975.

MIN,72 Minker, J., Wilson, G. A., Zimmerman, B. H., "An Evaluation of Query Expansion by the Addition of Clustered Terms for a Document Retrieval System". Information Storage and Retrieval, Vol. 8, pp. 329-348. Pergamon Press, Great Britain, 1972.

NCC,73 National Computing Center Ltd., "NCC Thesaurus of Computing Terms", Oxford Road, Manchester, England, 1973.

OCO,73 O'Connor, John, "Text Searching Retrieval of Answer - Sentences and Other Answer - Passages". Journal of the American Society for Information Science, 24:6 (November - December 1973) pp. 445-460.

PUL,65 Thesaurus of Pulp and Paper Terms, Point Claire, Quebec. Pulp and Paper Research Institute of Canada, 1965.

REA,77 Reader's Digest, "Family Word Finder".

ROC,66 J. J. Rochio, Jr., "Document Retrieval Systems - Optimization and Evaluation", Report No.: ISR-10 to the National Science Foundation, Department of Computer Science, Cornell University, 1966.

SAL,67 G. Salton, "Search Strategy and the Optimization of Retrieval Effectiveness", Report No.: ISR-12 to the National Science Foundation, Department of Computer Science, Cornell University, 1967.

SAL,68 G. Salton, "Automatic Information Organization and Retrieval", New York, McGraw - Hill Book Co., 1968.

SAL,68A Lesk, M. E., and Salton, G., "Evaluation of Interactive Search and Retrieval Methods Using Automatic Information Displays". Report No.: 68-17, Cornell University, Ithaca, N. Y., 1968.

SAL,72A G. Salton, "Experiments in Automatic Thesaurus Construction for Information Retrieval". Proceedings of IFIP Congress 1971, North Holland Publishing Co., Amsterdam, 1972.

SAL,72B G. Salton, "Dynamic Document Processing", Communications of the ACM, Vol. 15, No. 7, July 1972.

SAL,73A G. Salton, "Recent Studies in Automatic Text Analysis and Document Retrieval", Journal of the ACM., Vol. 20, NO. 2, April 1973.

SAL,73B G. Salton and C. S. Yang, "On the Specification of Term Values in Automatic Indexing", Journal of Documentation, 29 (no. 4), 1973.

SAL,73C G. Salton, "Comments on "An Evaluation of Query Expansion by the Addition of Clustered Terms for

a Document' Retrieval System"". Information Storage and Retrieval, Volume 8, p. 349. Pergamon Press, Great Britain , 1972.

SAL,75 G. Salton, A. Wong, C. S. Yang, "A Vector Space Model for Automatic Indexing", CACM, November 1975, Volume 18, No. 11.

SAL,77 G. Salton and C. T. Yu, "Effective Information Retrieval Using Term Accuracy". Communications of the ACM, Vol. 20, March 1977.

SCH,73 Schneider, J. H., Furth, S. B. (editors), "Survey of Commercially Available Computer Readable Bibliographic Data Bases". American Society for Information Science, Washington, D. C., 1973.

SPA,70 Sparck Jones, K., "Automatic Thesaurus Construction and the Relation of a Thesaurus to Indexing Terms". ASLIB Processing, Volume 22, Number 5 , page 228.

SPA,71A Sparck Jones, K., "Automatic Keyword Classification and Information Retrieval". London, England, Butterworths, 1971.

SPA,71B Sparck Jones, K. and Barber, E. O., "What Makes an Automatic Keyword Classification Effective ?". Journal of the American Society of Information Science, pp. 166-175, 1971.

SPAT,80 Helmuth Spath, "Cluster Analysis Algorithms for Data Reduction and Classification of objects". J. Wiley & Sons, 1980.

SUS,63 E. H. Susenguth Jr., "Use of Tree Structures for Processing Files". Communications of the ACM, Volume 6, Number 5, May 1963.

TES,69 Thesaurus of Engineering and Scientific Terms, New York, Engineers Joint Council, 1969.

THE,68 Claire K. Schultz, "Thesaurus of Information Science Terminology". Revised edition. Communication Service Corporation, Washington, D. C., 1968.

THI,72 Thiel, L. H. and Heaps, H. S., "Program Design for Retrospective Searches on Large Data Bases". Information Storage and Retrieval 8:1 (February, 1972).

UNE,70 UNESCO: "Guidelines for the Establishment and Development of Monolingual Scientific and Technical Thesauri for Information Retrieval". Paris, 1970.

VIC,66 B. C. Vickery, "Thesaurus - A New World in Documentation". Journal of Documentation, Volume 16, No. 4 pp. 181-189, Dec., 1966.

VIC,68 B. C. Vickery, "Faceted Classification: A Guide to Construction and Use of Specialized Schemes". Second edition, London: ASLIB, 1968.

WAT,66 Water Resources Thesaurus, Washington, D. C., United States Department of the Interior, Office of Water Resources Research, 1966.

WEL,73 M. Wells, "File Compression Using Variable Length Encodings". Computer Journal, 15:4 (November 1972) 308-313.

WON,73 A. Wong and C. S. Yang, "A Vector Space Model for Automatic Indexing", Technical Report no.: 74-208, Department of Computer Science, Cornell University, Ithaca, N. Y., 1974.

YU,74 C. T. Yu, "A Clustering Algorithm Based on User Queries", Journal of the Americal Society for Information Science, July 1974.

APPENDIX - A

Programs

```
C
C       PROGRAM SCANS ALL DOCUMENTS DESIGNATED BY THE SELECTED ID-S.
C       WORDS ARE EXTRACTED FROM TITLE, ABSTRACT AND KEY-WORD
C       PARAGRAPHS OF THE TEXT.
C
      PROGRAM GUY40(TFL,GASTOP,SELCR,OUTPUT,TAPE1=TFL,TAPE2=GASTOP
     +             ,TAPE3=SELCR,TAPE4,TAPE5=OUTPUT)
      DIMENSION EXT(8),TEXT(50,8),EXAM(71),TEMP(10),WORDS(1000),
     +          IFRQ(1000,4),TWORDS(1000),ITFRQ(1000,3),SPEC(9),
     +          STW(1900),CRS(500)
      REAL ID,NEW
      LOGICAL FLUSH,LINSW,LAST
      DATA BLANK,BLK1,BLK2/10H          ,6H    1 ,6H    2 /
      DATA EQUAL,SPEC/1H=,1H ,1H.,1H,,1H-,1H",1H(,1H),1H/,1H=/
      FLUSH=.TRUE.
      LINSW=.FALSE.
      LAST=.FALSE.
      LINE=0 $ IPR=0 $ IPT=0
      II=0
C
C       INITIAL SETUP
C
      READ(2,900)(STW(I),I=1,1848)
  900 FORMAT(8A10)
      READ(3,901)ICR
  901 FORMAT(I6)
      DO 5 I=1,ICR
      READ(3,902)A,CCAT
      IF(II.EQ.0)GO TO 3
      IF(A.EQ.CRS(II))GO TO 5
    3 II=II+1
      CRS(II)=A
    5 CONTINUE
      ICR=II
  902 FORMAT(A6,F4.2)
      I=1
      ID=CRS(I)
C
C     READ IN TEXT
C
   10 READ(1,903)NEW,(EXT(J),J=1,8)
  903 FORMAT(1X,A6,A3,7A10)
      IF(EOF(1))500,15
   15 IF(NEW.EQ.ID)GO TO 20
      IF(FLUSH)GO TO 10
      IF(NEW.NE.BLANK.AND.NEW.NE.BLK1.AND.NEW.NE.BLK2)GO TO 35
   20 FLUSH=.FALSE.
      LINE=LINE+1
      DO 25 J=1,8
   25 TEXT(LINE,J)=EXT(J)
      GO TO 10
C
C     PROCESS TEXT
C
   35 FLUSH=.TRUE.
      IPR=IPR+1
      DO 200 K=1,LINE
      REWIND 4
      WRITE(4,904)(TEXT(K,KK),KK=1,8)
  904 FORMAT(A3,7A10)
```

```
      IF(LEQUAL.GE.1)GO TO 45
   40 CONTINUE
      GO TO 200
   45 JLO=J+1
   46 IF(LINSW)47,50
   47 LINSW=.FALSE.
      GO TO 55
   50 JPT=0
      DO 52 IC=1,10
   52 TEMP(IC)=BLANK
   55 DO 60 JSC=JLO,71
      EX=BLANK
      EX=EXAM(JSC)
      DO 56 NN=1,9
   56 IF(EX.EQ.SPEC(NN))GO TO 70
      IF(LEQUAL.EQ.2)GO TO 60
      JPT=JPT+1
      IF(JPT.GT.10)GO TO 60
      TEMP(JPT)=EX
   60 CONTINUE
      IF(JPT.LT.10)GO TO 100
C
C  ASSEMBLE AND CHECK WORD IF IT IS A STOPWORD
C
   70 CONTINUE
      IF(EX.EQ.EQUAL)LEQUAL=LEQUAL+1
      IF(LEQUAL.GE.5)GO TO 210
      IF(JPT.EQ.0)GO TO 95
      ENCODE(10,906,WORD)TEMP
  906 FORMAT(10A1)
      DO 80 NN=1,1848
   80 IF(WORD.EQ.STW(NN))GO TO 95
      IFIL=1
      IF(LEQUAL.GE.3)IFIL=LEQUAL-1
      IF(KPT.EQ.0)GO TO 90
      DO 85 KK=1,KPT
      IF(TWORDS(KK).NE.WORD)GO TO 85
      ITFRQ(KK,IFIL)=ITFRQ(KK,IFIL)+1
      GO TO 95
   85 CONTINUE
   90 KPT=KPT+1
      TWORDS(KPT)=WORD
      ITFRQ(KPT,IFIL)=1
   95 JLO=JSC+1
      IF(JLO.GT.71)GO TO 200
      GO TO 50
  100 LINSW=.TRUE.
  200 CONTINUE
C
C   UPDATE WORD COUNTERS
C
  210 DO 240 KK=1,KPT
      IF(IPT.EQ.0)GO TO 225
      DO 220 JJ=1,IPT
      IF(WORDS(JJ).NE.TWORDS(KK))GO TO 220
      JL=0
      DO 215 JK=1,3
      IFRQ(JJ,JK)=IFRQ(JJ,JK)+ITFRQ(KK,JK)
  215 JL=JL+ITFRQ(KK,JK)
```

```
      DO 230 JK=1,3
  230 IFRQ(IPT,JK)=ITFRQ(KK,JK)
      IFRQ(IPT,4)=1
  240 CONTINUE
C
C     RESET VALUES
C
      DO 250 KK=1,LINE
      DO 250 JJ=1,8
  250 TEXT(KK,JJ)=BLANK
      DO 260 KK=1,KPT
      TWORDS(KK)=BLANK
      DO 260 JJ=1,3
      ITFRQ(KK,JJ)=0
  260 CONTINUE
      KPT=0
      LINE=0
      LEQUAL=0
      IF(LAST)510,400
  400 I=I+1
      IF(I.GT.ICR) GO TO 510
      ID=CRS(I)
      GO TO 15
C
C     PRINT RESULTS
C
  500 LAST=.TRUE.
      IF(.NOT.FLUSH)GO TO 35
  510 WRITE(5,910)ICR,IPR,IPT,CCAT
  910 FORMAT(10X,14HINPUT TEXT     ,I6/10X,14HPROCESSED TEXT,I6/
     +       10X,14HTOTAL NO WORDS,I6/10X,14HCATEGORY IS    ,F4.2///)
      DO 530 I=1,IPT
      ITFRQ(I,3)=I
      DO 530 J=1,3
      ITFRQ(I,1)=ITFRQ(I,1)+IFRQ(I,J)
  530 CONTINUE
  540 DO 550 I=2,IPT
      II=I-1
      IF(ITFRQ(ITFRQ(II,3),1).GE.ITFRQ(ITFRQ(I,3),1))GO TO 550
      NPTR=ITFRQ(II,3)
      ITFRQ(II,3)=ITFRQ(I,3)
      ITFRQ(I,3)=NPTR
      LSW=1
  550 CONTINUE
      IF(LSW.EQ.0)GO TO 560
      LSW=0
      GO TO 540
  560 REWIND 4
      WRITE(5,911)
  911 FORMAT(2(8X,4HWORD,9X,4HTITL,6X,4HABST,7X,3HKEY,6X,5HDB FR,
     +       5X,3HTOT,1X)//)
      WRITE(5,912)(WORDS(ITFRQ(I,3)),(IFRQ(ITFRQ(I,3),J),J=1,4),
     +             ITFRQ(ITFRQ(I,3),1),I=1,IPT)
      WRITE(4,913)CCAT
  912 FORMAT(2(5X,A10,5(4X,I6)))
  913 FORMAT(5X,F4.2)
      WRITE(4,912)(WORDS(ITFRQ(I,3)),(IFRQ(ITFRQ(I,3),J),J=1,4),
     +             ITFRQ(ITFRQ(I,3),1),I=1,IPT)
      STOP
```

```
C
C       FROM THE MAIN POOL, PROGRAM SELECTS AND SORTS THE
C       CR CATEGORIES REQUIRED FOR PROCESSING
C
        PROGRAM GUY38(CRSORT,TAPE1=CRSORT,TAPE2,OUTPUT,TAPE3=OUTPUT)
        DIMENSION TCR(500),CR(500)
        IPT=0 $ ISW=0
        SCR=4.40
    1   READ(1,900)AID,ACR
  900   FORMAT(A6,F4.2)
        IF(EOF(1))50,5
    5   IF(ACR.EQ.SCR) GO TO 10
        IF(ACR.GE.4.40.AND.ACR.LE.4.49)GO TO 10
        GO TO 1
   10   IPT=IPT+1
        TCR(IPT)=AID
        CR(IPT)=ACR
        GO TO 1
   50   DO 100 I=2,IPT
        J=I-1
        IF(TCR(J).LE.TCR(I))GO TO 100
        AID=TCR(J)
        ACR=CR(J)
        TCR(J)=TCR(I)
        CR(J)=CR(I)
        TCR(I)=AID
        CR(I)=ACR
        ISW=1
  100   CONTINUE
        IF(ISW.EQ.0) GO TO 150
        ISW=0
        GO TO 50
  150   WRITE(2,903)IPT
  903   FORMAT(I6)
        DO 170 I=1,IPT
  170   WRITE(2,900) TCR(I),CR(I)
        WRITE(3,901) IPT
  901   FORMAT(6HOUTPUT,I6)
        STOP
        END
 EOI ENCOUNTERED.
```

```
C
C       THIS PROGRAM SORTS THE ORIGINAL CR-S BY YEAR
C
        PROGRAM GUY33(UCR,TAPE1=UCR,TAPE2,OUTPUT,TAPE3=OUTPUT)
        DIMENSION TCR(1500),ICR(1500)
        IPT=1
        ISW=0
    1   READ(1,500)TCR(IPT),ICR(IPT)
  500   FORMAT(A6,F4.2)
        IF(EOF(1))50,10
   10   IPT=IPT+1
        GO TO 1
   50   IHI=IPT-1
        WRITE(3,501)IHI
  501   FORMAT(5HINPUT,I6)
   55   DO 100 I=2,IHI
        J=I-1
        IF(ICR(J).LE.ICR(I))GO TO 100
        CR=TCR(J)
        ICR1=ICR(J)
        TCR(J)=TCR(I)
        ICR(J)=ICR(I)
        TCR(I)=CR
        ICR(I)=ICR1
        ISW=1
  100   CONTINUE
        IF(ISW.EQ.0)GO TO 150
        ISW=0
        GO TO 55
  150   IPT=0
        DO 170 I=1,IHI
        WRITE(2,500)TCR(I),ICR(I)
        IPT=IPT+1
  170   CONTINUE
        WRITE(3,502)IPT
  502   FORMAT(6HOUTPUT,I6)
        STOP
        END
EOI ENCOUNTERED.
```

```
C       THIS PROGRAM SORTS AND PRINTS THE INPUT BY
C            T,TA,TK,A,AK,DB-FREQ.,ALPHA
C       WHERE: T=TITLE, A=ABSTRACT, K=KEY-WORD
C
        PROGRAM GUY50(TE1,OUTPUT,TAPE1=TE1,TAPE2=OUTPUT)
        DIMENSION WORDS(1200),IFR(1200,5),IMK(1200)
C
C
        JPT=975
        READ(1,902)CR
  902   FORMAT(5X,F5.2)
        READ(1,900)(WORDS(K),(IFR(K,KJ),KJ=1,5),K=1,JPT)
  900   FORMAT(2(5X,A10,5(4X,I6)))
        K=0
        JSW=1
        ISW=1
        DO 200 IK=1,4
  10    I=IK
        DO 20 J=1,JPT
  20    IMK(J)=J
  30    DO 50 J=2,JPT
        JJ=J-1
        IF(JSW.EQ.0)GO TO 45
        IF(IFR(IMK(JJ),I).GE.IFR(IMK(J),I))GO TO 50
        GO TO 47
   45   CONTINUE
        IF(IFR(IMK(JJ),IK).GT.IFR(IMK(J),IK))GO TO 50
        IF(IFR(IMK(JJ),I).GE.IFR(IMK(J),I))GO TO 50
  47    NPTR=IMK(JJ)
        IMK(JJ)=IMK(J)
        IMK(J)=NPTR
        LSW=1
  50    CONTINUE
        IF(LSW.EQ.0)GO TO 60
        LSW=0
        GO TO 30
  60    CONTINUE
        IF(ISW.EQ.1)GO TO 80
        I=K
        ISW=1
        JSW=0
        GO TO 30
  80    WRITE(2,901) CR
  901   FORMAT(1H1,10X,10HCATEGORY :,F5.2///
     +     2(8X,4HWORD,9X,4HTITL,6X,4HABST,7X,3HKEY
     +       6X,5HDB FR,5X,3HTOT,1X)///)
        WRITE(2,900)(WORDS(IMK(J)),(IFR(IMK(J),KJ),KJ=1,5),J=1,JPT)
        JSW=1
        IF(IK.GT.3)GO TO 200
        K=I+1
        IF(K.GT.3)GO TO 200
        I=K
        IF(IK.EQ.1.AND.K.EQ.3)GO TO 70
        JSW=0
        GO TO 30
  70    ISW=0
        GO TO 10
  200   CONTINUE
C
```

```
      JJ=J-1
      IF(WORDS(IMK(JJ)).LE.WORDS(IMK(J)))GO TO 250
      NPTR=IMK(JJ)
      IMK(JJ)=IMK(J)
      IMK(J)=NPTR
      LSW=1
250   CONTINUE
      IF(LSW.EQ.0)GO TO 260
      LSW=0
      GO TO 220
260   WRITE(2,901)CR
      WRITE(2,900)(WORDS(IMK(J)),(IFR(IMK(J),KJ),KJ=1,5),J=1,JPT)
      STOP
      END
EOI ENCOUNTERED.
```

```
C
C       FROM THE TEXT FILE OF THE CACM THIS PROGRAM CREATES ONE
C       OUTPUT ENTRY FOR EACH CR CATEGORY IN EACH DOCUMENT
C
      PROGRAM GUY30(TFL,OUTPUT,TAPE1=TFL,TAPE2,TAPE3=OUTPUT,TAPE4)
      DIMENSION TEXT(100,8),EXT(8),EXAM(73),CR(50),TCR(200,2),TEMP(4)
      DIMENSION NUM(11),MPTR(200)
      REAL NEW,IOLD,NUM
      LOGICAL FIRST,LAST,RSW,LINSW
      DATA STARS,BLANK/10H    ******,10H          /,EQUAL/1H=/
      DATA BLK1,BLK2/6H     1 ,6H     2 /
      DATA NUM/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H0,1H./
      LINSW=.FALSE.
      FIRST=.TRUE.
      LAST=.FALSE.
      RSW=.FALSE.
      LINE=0 $KPT=0 $IPT=0 $ITOT=0
      IREJ=0 $JOUTS=0 $IOUT=0 $ LEQUAL=0
C
C       READ IN ONE DOCUMENT
C
 1    READ(1,900)NEW,(EXT(J),J=1,8)
 900  FORMAT(1X,A6,A3,7A10)
      IF(EOF(1))500,10
 10   IF(.NOT.FIRST)GO TO 15
      FIRST=.FALSE.
      GO TO 20
 15   IF(NEW.EQ.IOLD.OR.NEW.EQ.BLANK)GO TO 20
      IF(NEW.EQ.BLK1.OR.NEW.EQ.BLK2)GO TO 25
      GO TO 100
 20   IF(NEW.NE.BLANK)IOLD=NEW
 25   LINE=LINE+1
      IF(LINE.GE.101) GO TO 510
      DO 30 J=1,8
      TEXT(LINE,J)=EXT(J)
 30   CONTINUE
      GO TO 1
C
C       EXAMINE ONE LINE AT A TIME
C
 100  ITOT=ITOT+1
      IF(RSW)101,104
 101  IREJ=IREJ+1
      IF(LAST)510,102
 102  RSW=.FALSE.
      GO TO 391
 104  DO 190 K=1,LINE
      REWIND 4
      WRITE(4,920)(TEXT(K,KK),KK=1,8)
 920  FORMAT(A3,7A10)
      REWIND 4
      READ(4,908)(EXAM(KK),KK=1,73)
 908  FORMAT(73A1)
C
C       SCAN FOR DELIMETER
C
      DO 105 J=1,73
      IF(EXAM(J).EQ.EQUAL)LEQUAL=LEQUAL+1
      IF(LEQUAL.GE.2)GO TO 110
 105  CONTINUE
```

```
      GO TO 125
  120 IF(JHI.GT.73)JHI=73
      JPT=0
      DO 122 IC=1,4
  122 TEMP(IC)=BLANK
  125 DO 130 JSC=JLO,JHI
      IF(EXAM(JSC).EQ.BLANK) GO TO 130
      IF(EXAM(JSC).EQ.EQUAL)GO TO 135
      IF(JPT.EQ.4)GO TO 135
      DO 127 NN=1,11
  127 IF(EXAM(JSC).EQ.NUM(NN))GO TO 129
      WRITE(3,933)K,JSC,EXAM(JSC)
  933  FORMAT(2I6,A10)
      GO TO 101
C
C      EXTRACT CR CATEGORIES
C
  129 JPT=JPT+1
      TEMP(JPT)=EXAM(JSC)
  130 CONTINUE
      IF(JPT.EQ.4)GO TO 135
      DO 132 JKK=JHI,73
  132 IF(EXAM(JKK).NE.BLANK)GO TO 135
      GO TO 140
  135 IPT=IPT+1
      ENCODE(4,902,CRM)TEMP
  902 FORMAT(4A1)
      CR(IPT)=CRM
      IF(EXAM(JSC).EQ.EQUAL)GO TO 200
      JLO=JHI+1
      JHI=JHI+5
      IF(JLO.GT.73)GO TO 190
      GO TO 120
  140 LINSW=.TRUE.
  190 CONTINUE
C
C          UPDATE CR TOTALS
C
  200 IF(IPT.EQ.0)GO TO 101
      REWIND 4
      DO 280 JK=1,IPT
  280 WRITE(4,980)CR(JK)
  980 FORMAT(A4)
      REWIND 4
      DO 290 JK=1,IPT
  290 READ(4,981)CR(JK)
  981 FORMAT(F4.2)
      DO 220 J=1,IPT
      IF(KPT.EQ.0)GO TO 215
      DO 210 KK=1,KPT
      IF(CR(J).NE.TCR(KK,1)) GO TO 210
      TCR(KK,2)=TCR(KK,2)+1.
      GO TO 220
  210 CONTINUE
  215 KPT=KPT+1
      IF(KPT.GE.201) GO TO 510
      TCR(KPT,1)=CR(J)
      TCR(KPT,2)=1.
C
```

```
 390  CONTINUE
      IF(LAST)510,395
C
C       WRITE REJECT TEXT
C
 391  WRITE(3,922)ITOT,IOLD,(TEXT(1,KK),KK=1,8)
 922  FORMAT(7HREC NO:,I6,2X,A6,A3,7A10)
      GO TO 395
      DO 330 KK=2,LINE
 330  WRITE(3,921)(TEXT(KK,L),L=1,8)
 921  FORMAT(21X,A3,7A10)
C
C          RESET VALUES
C
 395  CONTINUE
      DO 400 KK=1,LINE
      DO 400 JJ=1,8
 400  TEXT(KK,JJ)=BLANK
      DO 410 KK=1,IPT
 410  CR(KK)=BLANK
      LINE=0
      IPT=0
      LEQUAL=0
C
C          REJECT TEXT
C
      REJ=EXT(2).AND.000000007777777777777B
      IF(REJ.NE.STARS) GO TO 20
      RSW=.TRUE.
      GO TO 20
C
C            LAST TIME
C
 500  LAST=.TRUE.
      GO TO 100
 510  WRITE(3,940)ITOT,IOUT,IREJ,IOUTS,KPT
 940  FORMAT(10X,6HINPUT ,I6/10X,6HOUTPUT,I6/
     +              10X,6HREJECT,I6/10X,6HSPLIT ,I6/10X,8HCATEGORY,I6)
      ITOT=0
      DO 511 I=1,KPT
 511  ITOT=ITOT+TCR(I,2)
      DO 515 I=1,KPT
 515  MPTR(I)=I
 516  DO 518 I=2,KPT
      II=I-1
      IF(TCR(MPTR(II),2).GE.TCR(MPTR(I),2))GO TO 518
      NPTR=MPTR(II)
      MPTR(II)=MPTR(I)
      MPTR(I)=NPTR
      LSW=1
 518  CONTINUE
      IF(LSW.EQ.0)GO TO 520
      LSW=0
      GO TO 516
 520  CONTINUE
      WRITE(3,945)((TCR(MPTR(I),J),J=1,2),I=1,KPT)
 945  FORMAT(4(10X,F4.2,F8.0,3X))
      WRITE(3,950) ITOT
 950  FORMAT(10X,10HCAT. TOTAL,I8)
      STOP
```

```
C
C       THE PROGRAM CONSOLIDATES LOW LEVEL CR CATEGORY WORD FILES
C       INTO MID AND HIGH LEVEL CATEGORY FILES
C
      PROGRAM GUY45(OUTPUT,TE1,TE2,TE3,TE4,TE7,TAPE1=TE1,TAPE2=TE2,
     +          TAPE3=TE3,TAPE5=OUTPUT,TAPE6,TAPE4=TE4,TAPE7=TE7)
      DIMENSION WORDS(1000),IFR(1000,5),OUT(1400),IOUT(1400,2),
     +             CR(4),IPT(4),IMK(1400)
      DATA IPT/250,250,250,250/
      JPT=0
C
C       READ ALL INPUT FILES INTO "OUT"
C
      READ(7,913)JPT
      READ(7,912)(OUT(I),(IOUT(I,J),J=1,2),I=1,JPT)
    1 DO 200 I=1,4
      KK=IPT(I)
      READ(I,900)CR(I)
  900 FORMAT(5X,F4.2)
      READ(I,901)(WORDS(K),(IFR(K,KJ),KJ=1,5),K=1,KK)
  901 FORMAT(2(5X,A10,5(4X,I6)))
      DO 50 K=1,KK
      IF(JPT.EQ.0)GO TO 25
      DO 10 J=1,JPT
      IF(OUT(J).NE.WORDS(K))GO TO 10
      IOUT(J,1)=IOUT(J,1)+IFR(K,4)
      IOUT(J,2)=IOUT(J,2)+IFR(K,5)
      GO TO 50
   10 CONTINUE
   25 JPT=JPT+1
      IF(JPT.GT.1400)GO TO 500
      OUT(JPT)=WORDS(K)
      IOUT(JPT,1)=IFR(K,4)
      IOUT(JPT,2)=IFR(K,5)
   50 CONTINUE
  200 CONTINUE
C
C       POINTER SORT ALL DATA IN OUT
C
  500 DO 510 I=1,1400
  510 IMK(I)=I
      IF(JPT.GT.1400)JPT=1400
  520 DO 550 I=2,JPT
      II=I-1
      IF(IOUT(IMK(II),2).GE.IOUT(IMK(I),2))GO TO 550
      NPTR=IMK(II)
      IMK(II)=IMK(I)
      IMK(I)=NPTR
      LSW=1
  550 CONTINUE
      IF(LSW.EQ.0)GO TO 560
      LSW=0
      GO TO 520
  560 I=0
      DO 570 K=1,4
  570 I=I+IPT(K)
```

```
      REWIND 6
      WRITE(6,913)JPT
  913 FORMAT(I6)
      WRITE(5,910)(CR(II),II=1,4),I,JPT
  910 FORMAT(10X,12HCATEGORIES :,4F5.2/
     +       10X,12HINPUT WORDS ,I5/
     +       10X,12HOUTPUT WORDS,I5///)
      WRITE(5,911)
  911 FORMAT(4(8X,4HWORD,7X,5HDB FR,5X,3HTOT)//)
      WRITE(5,912)(OUT(IMK(I)),(IOUT(IMK(I),J),J=1,2),I=1,JPT)
      WRITE(6,912)(OUT(IMK(I)),(IOUT(IMK(I),J),J=1,2),I=1,JPT)
  912 FORMAT(4(5X,A10,5X,I3,5X,I4))
      STOP
      END
EOI ENCOUNTERED.
```

APPENDIX - B

High level selections

*CATEGORIES : 4.10 4.20 4.30 4.40
INPUT WORDS 1000
OUTPUT WORDS 758

WORD	DB FR	TOT	WORD	DB FR	TOT	WORD	DB FR	TOT	WORD	DB FR	TOT
PROGRAMMIN	128	297	COMPUTER	130	253	STORAGE	75	197	ALGORITHM	83	159
MULTIPROGR	68	141	LANGUAGES	78	138	PROCESSING	85	133	SHARING	53	119
SOFTWARE	50	118	IMPLEMENTA	77	113	INFORMATIO	63	110	PROCESSES	46	110
USER	50	107	PAGING	50	103	SCHEDULING	36	96	STRUCTURES	50	85
DISPLAY	23	84	VIRTUAL	45	84	ALGORITHMS	47	82	INPUT	43	76
SYNTAX	47	75	PERFORMANC	33	75	INDEPENDEN	38	74	METHOD	46	73
PARALLEL	32	72	COMPILER	38	70	ALLOCATION	35	70	IMPLEMENTE	64	67
DEFINITION	35	63	ENVIRONMEN	48	63	COMMUNICAT	39	62	SIMULATION	21	61
COMPUTATIO	44	58	OUTPUT	41	58	CONTEXT	28	57	GRAMMARS	13	56
INSTRUCTIO	24	56	MEASUREMEN	22	55	MATRIX	16	53	CRITERIA	21	53
TABLES	12	52	FUNCTIONS	30	51	REPRESENTA	29	51	HARDWARE	36	51
COLLECTION	30	50	MANIPULATI	28	49	PROCESSOR	31	48	EXECUTION	31	48
POSSIBLE	40	47	APPLICATIO	39	46	ACCESS	25	46	NETWORK	14	44
GARBAGE	15	44	TRANSLATOR	20	43	DESCRIPTIO	42	43	GRAPHIC	12	42
ARRAY	9	41	ORGANIZATI	24	41	COMPUTING	23	41	DYNAMIC	26	41
LISP	18	40	FORTRAN	25	40	EFFICIENT	35	38	COMPUTERS	28	37
SUPERVISOR	12	37	SEMANTICS	27	36	CODE	20	36	INTERACTIV	18	36
OVERLAY	7	36	PROCESSORS	17	36	FORMAL	25	35	EXTENDED	27	35
AUTOMATIC	16	35	EDITING	9	34	PARSING	15	33	ALGOL	17	33
PARTITIONI	7	33	ENTRY	14	33	SPECIFICAT	25	33	DEBUGGING	15	33
SEGMENTATI	18	33	MACRO	19	32	COMPILERS	24	32	VARIABLES	21	32
EFFICIENCY	29	32	TEXT	18	32	DEFINED	27	32	EXAMPLES	32	32
SCHEME	19	32	CHARACTERI	27	32	ORIENTED	19	31	FACILITY	19	31
RESOURCE	16	31	EQUATIONS	7	30	PARSE	5	29	CONSTRUCTI	22	29
STRING	16	28	OPTIMIZATI	13	28	PLOTS	4	28	EXTENSION	22	27
EVALUATION	22	27	INTERRUPT	9	27	GRAPHICS	15	27	FOLDING	5	26
SORTING	8	26	PL	8	25	BLOCKS	9	25	EXISTING	19	25
IBM	18	25	TERMINAL	18	25	GENERATING	13	24	PARSER	10	23
ILLUSTRATE	22	23	PROGRAMMER	18	23	MACHINES	21	23	CONVENTION	20	23
ADDRESSING	9	22	CALCULATIO	11	22	GENERATION	18	22	DEVELOPMEN	16	22
AFFIX	3	21	PRECEDENCE	3	21	TRANSFORMA	14	21	MANUAL	11	21
SUBEXPRESS	5	21	PERMITS	19	21	MODULES	8	21	PAGED	15	21
MULTIPLE	12	21	INTERFEREN	8	21	MULTIPROCE	18	21	GRAMMAR	10	20
SYMBOLIC	14	20	COMPILATIO	17	20	MATRICES	12	20	SPECIFIED	17	20
CONVERSION	13	20	CIRCUIT	5	20	LOGICAL	17	20	COMPACT	6	20
DOCUMENTAT	4	20	CORRECTNES	13	20	CONCEPTS	16	20	HIERARCHY	9	20
MODELS	17	20	SYNTACTIC	16	19	NETWORKS	6	19	SEQUENCES	10	19
COORDINATE	6	19	CAPABILITY	18	19	PROOFS	14	19	TRANSITION	8	19
INTERACTIO	14	19	CONFIGURAT	14	19	CONCURRENT	12	19	SYNCHRONIZ	12	19
CORE	12	19	MULTICS	7	19	CONVENIENT	13	18	CHECKING	6	18
INCREMENTA	8	18	SYMBOL	13	18	CPU	6	18	USERS	16	18
REFERENCE	13	17	GRAPHS	9	17	CODING	17	17	PARTIAL	10	17
DIGITAL	8	17	LINEAR	9	17	MATHEMATIC	11	17	MUTUAL	7	17
SHARED	12	17	ARRAYS	6	16	DEFINE	12	16	OPTIMUM	3	16
GENERATOR	11	16	DRIVEN	6	16	LISTS	12	16	OUTLINED	12	16
LINC	4	16	FILING	4	16	AMESPLOT	4	16	SEQUENTIAL	9	16

GRAPHICAL	3	13	MONITOR	7	13	INTERPRETE	11	12	APPLE	7	12
TRANSLATIO	5	12	GRASPE	2	12	OPTIMAL	7	12	ADVANTAGES	11	12
DIMENSIONA	6	12	CAPABILITI	9	12	AIDED	3	12	OVERLAYS	3	12
RELATIVE	10	12	PROOF	6	12	LINKAGE	9	12	LAP6	4	12
SCROLL	4	12	SUBPLOTS	4	12	SPELLING	2	12	PLEX	4	12
ESTIMATION	3	12	PL360	4	12	PARAMETERS	4	12	ASSEMBLER	8	11
INDEXING	3	11	GRAPH	4	11	NOTION	6	11	INTRODUCED	11	11
COMPACTING	5	11	CORRECT	5	11	SCATTER	8	11	CDC	9	11
REPLACEMEN	10	11	BATCH	6	11	COMPLEMENT	1	11	ALGORITHMI	10	10
NOTATION	10	10	DERIVE	4	10	ASP	2	10	COMPOSITIO	4	10
DETERMINIS	6	10	ALTERNATIV	10	10	FACILITATE	4	10	ADVANCED	4	10
REPRODUCIB	5	10	PORTABLE	3	10	CASES	9	10	SLIP	2	10
FORMULATED	10	10	GOALS	5	10	STORED	9	10	PAGES	7	10
TENEX	2	10	PROVE	5	10	COOPERATIN	9	10	SOLUTIONS	7	10
RETRIEVAL	4	9	KOSTER	3	9	LEXICAL	2	9	ASSOCIATIV	3	9
MACROS	9	9	CONTENT	9	9	CONDITIONS	9	9	TRACE	3	9
EXECUTED	5	9	ORDERING	3	9	COMMANDS	7	9	MODIFICATI	6	9
REMOTE	5	9	COMBINATIO	9	9	ABSTRACTIO	7	9	MULTI	9	9
INSTRUMENT	6	9	DEADLOCK	5	9	MATCHING	4	8	ISSUES	2	8
EXTENSIONS	8	8	LR	3	8	CURVILINEA	2	8	STAGE2	2	8
EXTENSIBLE	4	8	INTERPRETI	8	8	LOADER	6	8	APPLICABLE	8	8
STRUCTURAL	6	8	IMPROVED	5	8	FEATURE	8	8	REPRESENTE	7	8
INCORPORAT	8	8	MAP	5	8	FRAMEWORK	8	8	SCOPE	7	8
ANALYZED	6	8	JOBS	6	8	EXAMINED	8	8	MEASURING	4	8
HASHING	4	8	COMPARED	8	8	FOURIER	1	8	SIMPLIFICA	1	8
BRAILLE	1	8	MERGING	3	8	RATIONAL	3	7	METACOMPIL	7	7
PATTERNS	5	7	COMPILING	6	7	AUTOMATA	3	7	BOOLEAN	3	7
MAD	3	7	LOGIC	3	7	68	3	7	MANIPULATE	7	7
BURROUGHS	7	7	B5500	6	7	THEORETICA	6	7	FREQUENCY	4	7
EXPERIENCE	6	7	CONVERSATI	5	7	EXECUTING	4	7	DESIGNING	6	7
RELIABLE	4	7	VARIATIONS	7	7	SEMIAUTOMA	2	7	OVERHEAD	6	7
PRIMITIVES	7	7	ARCHITECTU	3	7	MICROPROGR	3	7	WIRING	1	7
MERGE	2	7	PREPARATIO	4	6	COMPLETENE	5	6	FLOWCHART	1	6
LINGUISTIC	4	6	FORMAT	4	6	EMULATION	2	6	APPROACHIN	2	6
ASSEMBLY	6	6	GAN	3	6	BOUNDED	1	6	CONSTRUCTS	6	6
SUITABLE	6	6	NAVIER	2	6	CONTINUITY	2	6	IITRAN	2	6
CONSTRUCTE	5	6	CALCULI	4	6	RECOMPILAT	2	6	STRUCTURED	3	6
ARBITRARY	6	6	DESIGNER	4	6	DESCRIBE	5	6	AMBIGUITY	4	6
DIAGNOSTIC	6	6	FORMALISM	3	6	COMBINES	6	6	ROW	4	6
6000	4	6	ERROR	4	6	MODELING	4	6	EMBEDDED	5	6
REENTRANT	2	6	SIGNIFICAN	6	6	APPROACHES	6	6	6600	4	6
DISTRIBUTE	4	6	BALANCING	4	6	DEVICES	6	6	IMPROVEMEN	6	6
EFFICIENTL	4	6	INTERFACES	4	6	DECREASE	3	6	HUMAN	2	6
SEGMENTED	4	6	READERS	2	6	CONTROLLAB	2	6	SEGMENT	4	6
MECHANISMS	4	6	SEGMENTS	4	6	ALLOCATED	5	6	COMMENT	5	6
QUOTIENT	5	6	STEPWISE	3	6	SUBROUTINE	4	6	CREATION	4	6
DUPLEX	1	6	RASTER	3	6	POLYNOMIAL	1	6	DETERMINAN	1	6
NONRECURSI	3	5	TESTING	5	5	PROGRAMMED	5	5	RESTRICTED	5	5
SNOBOL	2	5	INDIRECT	2	5	DIFFERENTI	1	5	ANALYZING	5	5
RUNS	5	5	EXPLICIT	5	5	CONTRIBUTE	5	5	REQUIRES	5	5
HEURISTIC	3	5	EXERCISES	1	5	7094	3	5	HIERARCHIE	3	5
MEDIUM	3	5	COMBINED	4	5	PRIVACY	3	5	CONTROLLED	5	5
INTENDED	5	5	ESTABLISHE	5	5	DYNAMICALL	5	5	COMMENTS	3	5
TASKS	4	5	ALLOWS	5	5	TRANSFORM	1	5	CLUSTERING	3	5
DECOMPOSIT	2	5	CONTAINING	3	5	AUTOMATICA	5	5	DEVICE	5	5
TV	1	5	DISPLAYED	4	5	OBTAINING	3	5	CHARACTERS	2	5

TELETYPE	2	4	PROBABILIT	4	4	ELIMINATIO	2	4	EXTREMELY	1	4
IMAGES	2	4	MICROFILM	1	4	QUICKSORT	1	4	GENERALIZA	1	4
OPERATE	3	3	AIDS	3	3	CONFLICT	2	3	CORRESPOND	3	3
RECOGNIZER	2	3	SENTENCE	1	3	SUBSET	3	3	GENERATE	3	3
REGISTER	2	3	PARTITION	3	3	FACTORING	1	3	FACTORIZAT	1	3
DETECTION	3	3	VECTORS	3	3	KNUTH	2	3	REDUCTION	1	3
PARSERS	2	3	TRAC	1	3	TRULY	2	3	60	2	3
HANDLED	3	3	GENERALITY	3	3	DERIVING	3	3	SIMULATED	2	3
PROVING	2	3	INDUCTION	2	3	CONSISTS	3	3	FILES	1	3
DISCUSSES	3	3	CONCATENAT	3	3	PRIMITIVE	3	3	NAMING	1	3
ATTEMPTS	3	3	ATTEMPT	3	3	INVOLVES	3	3	CATEGORIES	3	3
EQUIVALENT	3	3	PHYSICS	2	3	GEDANKEN	2	3	TYPELESS	2	3
LABELS	3	3	LABEL	2	3	LAMBDA	3	3	INTEGRATED	2	3
EVALUATING	1	3	FINITENESS	1	3	URBAN	2	3	ANIMATOR	1	3
ANIMATION	1	3	CAI	1	3	CATALYST	1	3	NESTED	2	3
7090	1	3	SYSTEMS:	2	3	QUANTITIES	3	3	ESTIMATING	3	3
DIVIDED	2	3	ACCOMPLISH	3	3	FAULTS	2	3	EXECUTE	3	3
OVERALL	3	3	YIELDS	2	3	GRADING	1	3	BIOMEDICAL	1	3
KARYOTYPIN	1	3	STRUCTURIN	2	3	FILED	1	3	DIRECTORY	2	3
NORMALLY	2	3	TRADITIONA	3	3	FUTURE	3	3	MULTIACCES	2	3
INVESTIGAT	2	3	STRATEGIES	2	3	DEADLOCKS	2	3	PRODUCTIVI	2	3
PROCEEDING	1	3	SIZES	1	3	VARIABLE	3	3	DESCRIBES	3	3
EFFECTIVEN	2	3	SCHEDULER	1	3	B6500	1	3	CONTENTION	1	3
BARRIERS	1	3	SYMMETRY	3	3	PRECISION	2	3	DISCRETE	2	3
SPECTRUM	2	3	BIBLIOGRAP	1	3	CALCULATED	3	3	KEYS	3	3
PRESENTATI	1	3	LAYOUT	2	3	GENERALIZE	2	3	RADIX	2	3
ETCHING	1	3	MAZE	1	3	UNCONVENTI	1	3	TRANSMISSI	1	3
BRAD	2	3	BROOKHAVEN	2	3	DISPLAYS	2	3	EMPLOYED	3	3
BINARY	2	3	ROTATING	2	3	SWEPT	1	3	DIFFICULT	3	3
INEXPENSIV	3	3	BLIND	1	3	TACTILE	1	3	ONLINE	1	3
ILLUSTRATI	3	3	COLLINS	1	3	RESULTANT	1	3	SAMPLE	3	3
NUMERICAL	3	3	CONCAVE	1	3	TELEVISION	2	3	RESULTING	2	3
UNIFORM	2	3	BOUND	1	3	RATIO	2	3	BRANCHING	1	3
SATISFACTO	2	2	FREQUENCIE	2	2	VARIANT	2	2	TIMING	2	2
IDENTIFIER	2	2	SENTENCES	2	2	STACK	2	2	EFFECTIVEL	2	2
PURPOSES	2	2	LENGTHS	2	2	HANDLE	2	2	PERFORMING	2	2
DIFFICULTI	2	2	EMBEDDING	2	2	FOUNDATION	2	2	ASYNCHRONO	2	2
ACCOMMODAT	1	2	ACHIEVING	2	2	TRADEMARK	1	2	GPL	1	2
DECLARATIO	2	2	POINTERS	2	2	EXTENDING	2	2	MINIMIZES	2	2
POLISH	1	2	REVIEW	1	2	AUTOMATE	1	2	ASSIGNED	2	2
NON	1	2	LOADING	2	2	LINKING	1	2	MEMORIES	2	2
ADDRESSABL	2	2	EXECUTABLE	2	2	ARRANGEMEN	2	2	SLOWER	2	2
VECTOR	2	2	REFERENCES	2	2	GAINS	2	2	FLEXIBLE	2	2
ORGANIZED	2	2	PARALYSIS	1	2	OCCUR	2	2	AMMING	2	2
MAXIMIZE	2	2	SUBOPERATI	1	2	UTILIZE	1	2	FORMALIZAT	2	2
ANALYTIC	2	2	QUANTUM	2	2	NUMERICALL	2	2	GRAPHICALL	1	2
IMPLICATIO	2	2	MARKOVIAN	2	2	KNOWN	2	2	AUTHOR	2	2
FEELS	2	2	THEREBY	2	2	MTS	1	2	IDENTIFIED	1	2
EMPIRICAL	2	2	PERFORMED	2	2	ADEPT	1	2	TUTORIAL	2	2
CATALOGED	1	2	FLEXIBILIT	2	2	VALUED	1	2	INTEGRAL	2	2
SUBJECT	1	2	BRIEF	2	2	EXTENSIVE	2	2	CHAINED	1	2
EXPLORE	2	2	AXIOMS	1	2	THEOREM	2	2	SCHEMES	2	2
ELECTRICAL	1	2	INCREASE	2	2	NELSON	1	2	QUADRATIC	2	2
COMPARISON	2	2	POLYGONAL	1	2	WARNOCK	1	2	PLOTTING	1	2
ATTRIBUTES	2	2	SCALED	1	2	SCALING	1	2	INTERACT	2	2

CONTINUOUS	2	2	COMPUTE	2	2	POINTING	1	2	SCREEN	2	2
BRAILLER	1	2	AID	2	2	LOCATING	2	2	ACCEPTS	1	2
PRODUCES	1	2	PRESENTLY	2	2	UNSATISFAC	2	2	TYPING	1	2
PRIME	2	2	PROPORTION	2	2						

2.051 CP SECONDS EXECUTION TIME

APPENDIX - C

Mid level selections

1 TOTAL NO WORDS 869
CATEGORY IS 4.10

WORD	TITL	ABST	KEY	DB FR	TOT	WORD	TITL	ABST	KEY	DB FR	TOT
COMPUTER	3	18	13	18	34	PROGRAMMIN	1	16	8	11	25
STORAGE	1	11	9	11	21	TABLES	4	12	5	4	21
ALGORITHM	2	14	3	11	19	LANGUAGES	1	13	3	9	17
SYNTAX	1	6	10	8	17	COMPILER	0	8	7	7	15
TRANSLATOR	2	6	6	7	14	ALGORITHMS	1	11	2	5	14
SOFTWARE	2	9	3	6	14	VIRTUAL	2	5	6	7	13
LISP	2	5	6	5	13	GRAMMARS	1	10	1	2	12
USER	0	11	1	5	12	MULTIPROGR	1	5	6	4	12
PAGING	1	2	9	7	12	INSTRUCTIO	0	9	2	4	11
PROCESSING	0	3	8	8	11	CRITERIA	1	9	1	5	11
PROCESSES	1	7	3	4	11	ENTRY	2	8	1	4	11
OVERLAY	1	7	3	2	11	MANIPULATI	2	2	6	4	10
GARBAGE	1	5	4	5	10	AUTOMATIC	2	4	4	4	10
SCHEDULING	1	6	2	2	9	FOLDING	1	5	3	2	9
DOCUMENTAT	1	4	4	1	9	APPLICATIO	1	7	0	6	8
POSSIBLE	0	8	0	6	8	SEGMENTATI	1	4	3	4	8
IMPLEMENTA	1	6	1	5	8	STRING	0	4	4	5	8
EDITING	0	4	4	2	8	DISPLAY	0	3	5	3	8
MACRO	0	5	2	4	7	INDEPENDEN	0	5	2	2	7
COMPILERS	0	3	4	5	7	FORMAL	0	6	1	6	7
MANUAL	1	5	1	4	7	PARALLEL	0	4	3	2	7
INFORMATIO	0	4	3	5	7	ENVIRONMEN	3	3	1	5	7
COMPUTATIO	1	4	2	5	7	CONTEXT	0	5	2	5	7
NETWORK	0	7	0	1	7	CONVERSION	3	2	2	5	7
PARSE	1	6	0	1	7	TEXT	0	4	3	4	7
PARSING	0	4	3	2	7	INTERPRETE	0	3	4	6	7
FUNCTIONS	0	7	0	5	7	PLOTS	0	6	1	1	7
COMPACTING	2	2	3	3	7	AFFIX	1	6	0	1	7
MASK	2	2	3	1	7	PROCESSOR	2	1	3	6	6
PROCESSORS	0	5	1	4	6	EXTENSION	0	6	0	5	6
EXISTING	0	6	0	4	6	DESCRIPTIO	0	6	0	6	6
COMPUTING	1	5	0	4	6	INPUT	0	3	3	3	6
DEFINE	0	6	0	4	6	PROGRAMMER	0	5	1	4	6
GENERATING	2	4	0	3	6	COLLECTOR	1	2	3	4	6
OPTIMUM	0	6	0	1	6	STRUCTURES	1	3	1	4	5
EXECUTION	0	5	0	3	5	EVALUATION	0	4	1	4	5
IMPLEMENTE	0	5	0	5	5	CALCULATIO	0	5	0	3	5
SPECIFICAT	0	5	0	5	5	RETRIEVAL	1	3	1	2	5
DEFINITION	0	4	1	4	5	COLLECTION	0	3	2	5	5
PL	0	4	1	1	5	CIRCUIT	1	3	1	1	5
INTERACTIO	1	2	2	3	5	GRAPHIC	0	4	1	2	5
NONRECURSI	2	1	2	3	5	METHOD	1	3	1	3	5
CPU	1	3	1	1	5	EFFICIENCY	1	1	2	3	4
FORTRAN	0	4	0	2	4	SOPHISTICA	0	4	0	4	4
ALLOCATION	0	2	2	4	4	COMPUTERS	0	2	2	3	4
SYNTACTIC	0	3	1	3	4	CODE	1	3	0	2	4
SEMANTICS	0	1	3	4	4	ASSEMBLER	0	2	2	3	4

INTERPRETI	0	4	0	4	4	SIMULATION	0	1	3	3	4
ESTIMATION	0	2	2	1	4	SYMBOLIC	0	3	1	3	4
CONVENIENT	0	4	0	3	4	MATCHING	0	2	2	2	4
STRINGS	0	4	0	4	4	NETWORKS	0	3	1	1	4
EXTENDED	0	3	1	3	4	COMMUNICAT	1	2	1	3	4
REQUESTS	0	4	0	2	4	FORMULATED	0	4	0	4	4
STANDARDIZ	1	2	1	2	4	TRANSFORMA	0	2	2	3	4
INTERACTIV	1	3	0	2	4	CAPABILITI	0	4	0	3	4
PROPOSAL	1	3	0	2	4	EFFICIENT	1	3	0	4	4
PERFORMANC	0	2	2	3	4	ISSUES	0	4	0	1	4
ACCESS	1	2	1	3	4	PL360	1	2	1	1	4
LINC	0	3	1	1	4	FILING	0	2	2	1	4
ORIENTED	0	2	2	2	4	AIDED	1	2	1	1	4
AMESPLOT	1	3	0	1	4	HARDWARE	0	2	2	3	4
CONFIGURAT	0	4	0	3	4	MODULES	0	4	0	1	4
SCHEME	0	4	0	3	4	RECURSION	1	1	2	3	4
PARTIAL	0	4	0	3	4	ADDRESSING	0	2	2	1	4
MUTUAL	0	3	1	2	4	OVERLAYS	1	3	0	1	4
INTERFEREN	1	2	1	1	4	DRIVEN	1	2	1	1	4
CONSTRUCTI	0	3	0	2	3	FACILITY	0	3	0	3	3
EXTENSIONS	0	3	0	3	3	REFERENCE	0	2	1	3	3
PARSER	0	1	2	2	3	GENERATOR	0	1	2	3	3
DESIGNER	0	3	0	2	3	PAGED	1	1	1	2	3
DESCRIBE	0	3	0	3	3	FREQUENCY	0	3	0	2	3
DERIVED	0	2	1	2	3	UTILIZATIO	0	1	2	2	3
SYMBOL	0	1	2	2	3	ALGOL	0	3	0	1	3
HANDLING	0	2	1	3	3	RECURSIVE	0	1	2	3	3
PREPARATIO	0	3	0	2	3	FINITE	0	1	2	2	3
ALGORITHMI	0	1	2	3	3	AMBIGUITY	1	1	1	2	3
COMPLETENE	0	2	1	2	3	CONVENTION	0	3	0	3	3
DIAGNOSTIC	0	3	0	3	3	REMOTE	1	2	0	2	3
BRIEFLY	0	3	0	3	3	REQUEST	1	1	1	2	3
FLOWCHART	0	2	1	2	3	DEMONSTRAT	0	2	1	3	3
LINGUISTIC	0	1	2	2	3	FORMAT	0	2	1	2	3
GRAMMAR	0	1	2	2	3	LOADER	1	0	2	2	3
IBM	0	2	1	2	3	OPERATE	0	3	0	3	3
LINKAGE	0	0	3	2	3	APAREL	1	2	0	1	3
VARIABLES	0	3	0	3	3	MULTIPLE	0	3	0	2	3
SEQUENCES	0	3	0	2	3	CONDITIONS	0	3	0	3	3
TESTED	0	3	0	3	3	TRANSLATIO	1	1	1	1	3
EQUATIONS	1	2	0	1	3	PUSHDOWN	0	2	1	1	3
MACHINES	0	2	1	3	3	AIDS	0	3	0	3	3
LAP6	0	2	1	1	3	SCROLL	0	2	1	1	3
TAPE	0	1	2	1	3	OUTLINED	0	3	0	2	3
PERFORM	0	3	0	2	3	GRAPHICS	0	0	3	2	3
ORGANIZATI	0	1	2	3	3	PLOT	0	2	1	1	3
SELF	0	2	1	1	3	SUBPLOTS	0	3	0	1	3
FORMALISM	1	2	0	1	3	BLOCKS	0	3	0	1	3
APPLICABLE	0	3	0	3	3	PROOFS	0	2	1	2	3
CORRECTNES	0	2	1	2	3	RATIONAL	1	1	1	1	3
EXCLUSION	0	2	1	1	3	TRANSITION	0	3	0	1	3
KOSTER	0	3	0	1	3	EMULATION	1	1	1	1	3
OPTIMAL	1	1	1	1	3	APPROACHIN	0	3	0	1	3
TRACE	1	1	1	1	3	SATISFACTO	0	2	0	2	2
NOTATION	0	2	0	2	2	LOGICAL	0	2	0	2	2

MACROS	0	2	0	2	2	CODING	0	2	0	2	2
IMPROVED	1	1	0	1	2	HASH	1	1	0	1	2
SCATTER	1	1	0	1	2	SEMANTIC	0	2	0	2	2
METACOMPIL	0	1	1	2	2	META	0	0	2	2	2
REPRODUCIB	0	1	1	1	2	CONTENT	0	2	0	2	2
SCOPE	0	2	0	2	2	NOTION	0	2	0	1	2
EXAMPLES	0	2	0	2	2	DETERMINIS	0	0	2	1	2
TESTING	0	1	1	2	2	EXECUTING	0	2	0	1	2
REQUIRES	0	2	0	2	2	CORE	0	2	0	2	2
REPLACEMEN	0	1	1	2	2	DYNAMIC	0	1	1	2	2
FREQUENCIE	1	1	0	2	2	JUMPS	1	1	0	1	2
CASES	0	2	0	2	2	EASIER	0	1	1	1	2
ESTIMATE	0	1	1	1	2	SUM	0	2	0	2	2
VARIANT	0	1	1	2	2	DETAILS	0	2	0	2	2
TIMING	1	1	0	2	2	PROGRAMMED	0	2	0	2	2
COMMANDS	0	2	0	2	2	CONDITIONA	0	1	1	1	2
ALTERNATIV	0	2	0	2	2	PATTERNS	0	2	0	2	2
QUANTIFIER	0	1	1	1	2	IDENTIFIER	0	2	0	2	2
BURROUGHS	0	2	0	2	2	B5500	0	2	0	2	2
NEWORK	0	2	0	1	2	NEWROK	0	2	0	1	2
CHECKING	0	2	0	1	2	AUTOMATICA	0	2	0	2	2
DEFINED	0	2	0	2	2	ESTABLISHE	0	2	0	2	2
TERMINAL	0	2	0	2	2	INCREMENTA	0	1	1	1	2
TYPEWRITER	1	1	0	1	2	CONSOLE	1	1	0	1	2
DOCUMENTS	0	2	0	1	2	EQUIVALENT	0	2	0	2	2
ILLUSTRATI	0	1	1	2	2	TESTER	1	0	1	1	2
LINGUIST	0	2	0	1	2	PHONOLOGY	0	1	1	1	2
FRAMEWORK	0	2	0	2	2	PHONEMES	0	2	0	1	2
SPECIFIED	0	2	0	2	2	APPLYING	0	2	0	2	2
EXECUTED	0	2	0	2	2	CDC	0	1	1	1	2
6000	0	1	1	1	2	GE	0	1	1	1	2
635	0	1	1	1	2	360	0	1	1	1	2
UNIVAC	0	1	1	1	2	1108	0	1	1	1	2
EXPERIENCE	0	2	0	2	2	OCCUPATION	0	1	1	1	2
COMPACTION	1	1	0	2	2	SUBROUTINE	0	2	0	2	2
PREPROCESS	0	1	1	2	2	NELSON	0	2	0	1	2
MECHANISM	0	2	0	1	2	PERMITTING	0	2	0	2	2
COMPARED	0	2	0	2	2	INTERFACE	0	2	0	2	2
EXAMINED	0	2	0	2	2	ARISE	0	2	0	2	2
CONCLUDED	0	2	0	2	2	HIERARCHIE	0	0	2	1	2
CYCLIC	1	0	1	1	2	PORTIONS	0	2	0	1	2
EXECUTABLE	0	2	0	2	2	TRANSFERS	0	2	0	2	2
SUBJECT	0	2	0	1	2	RAMAMOORTH	0	2	0	1	2
LIMITATION	0	2	0	2	2	SEGMENT	0	1	1	2	2
ASSEMBLERS	0	1	1	2	2	TRANSDUCTI	0	1	1	1	2
DERIVING	0	2	0	2	2	AUTOMATA	0	1	1	1	2
PRIORITY	0	2	0	2	2	MACHINE;	0	2	0	1	2
MANUFACTUR	0	2	0	2	2	ABLE	1	1	0	2	2
CONSTRUCTE	0	2	0	2	2	PHRASE	0	2	0	1	2
RESULTING	0	2	0	1	2	FORMATION	0	2	0	2	2
INDEXED	0	1	1	1	2	STRUCTURED	0	0	2	2	2
CONVERSATI	1	0	1	1	2	2048	1	1	0	1	2
ADDING	0	2	0	2	2	TAPES	0	2	0	1	2
DISK	1	1	0	2	2	MONITOR	0	1	1	1	2
PLOTTING	1	0	1	1	2	EXTENSIBLE	1	1	0	2	2

SCALED	0	1	1	1	2	MAP	0	1	1	1	2
SCALING	0	2	0	1	2	STRUCTURIN	0	2	0	2	2
DEVICE	1	0	1	2	2	BOOTSTRAPP	0	1	1	1	2
COMPACT	0	1	1	1	2	COMMENT	2	0	0	2	2
ADDRESSES	1	1	0	1	2	RELATIVE	0	2	0	2	2
COMPILING	0	1	1	2	2	COVERING	0	1	1	1	2
INTEGER	0	1	1	2	2	INDUCTION	1	0	1	1	2
PROVING	0	2	0	1	2	ALTRAN	1	1	0	1	2
EASE	0	2	0	2	2	CONSIDERAB	0	2	0	2	2
CHENEY	2	0	0	2	2	ALGORITH	1	0	1	2	2
WRITABLE	0	2	0	1	2	SOLUTIONS	0	2	0	1	2
SUBPROBLEM	0	2	0	1	2	PROVE	0	2	0	1	2
WIJNGAARDE	0	2	0	1	2	FLOYD	0	1	1	1	2
ILLUSTRATE	0	2	0	1	2	MICROPROGR	0	1	1	1	2
MODELS	0	0	2	2	2	FLOWCHARTS	1	1	0	1	2
FINDING	0	2	0	1	2	MODELING	1	1	0	1	2
IMPLEMENTI	0	1	0	1	1	CONTENDED	0	1	0	1	1
FUNCTIONAL	0	1	0	1	1	CONSTITUES	0	1	0	1	1
CANONICAL	0	1	0	1	1	SUPPORTING	0	1	0	1	1
CONFLICT	0	1	0	1	1	DIFFICULTI	0	1	0	1	1
UNIVERSAL	0	1	0	1	1	METALANGUA	0	1	0	1	1
METAPHYSIC	0	1	0	1	1	EMBEDDING	0	1	0	1	1
FALSE	0	1	0	1	1	ECONOMICS	0	1	0	1	1
FORTRANAUT	0	1	0	1	1	SURVEY	0	1	0	1	1
FOUNDATION	0	1	0	1	1	ALTEXT	0	1	0	1	1
REVIEW	0	1	0	1	1	AUTOMATE	0	1	0	1	1
APPROACHES	0	1	0	1	1	AUTOMATING	0	1	0	1	1
POSTSYNTAC	0	1	0	1	1	TOPICS	0	1	0	1	1
MACROPROCE	0	0	1	1	1	DESIGNING	1	0	0	1	1
FACIL	1	0	0	1	1	ADOPT	0	1	0	1	1
EXPLICIT	0	1	0	1	1	ACCEPTING	0	1	0	1	1
REJECTING	0	1	0	1	1	RECORDABIL	0	1	0	1	1
ASYNCHRONO	0	1	0	1	1	IMPLY	0	1	0	1	1
DESIRES	0	1	0	1	1	INFLUENCES	0	1	0	1	1
AFFECTING	0	1	0	1	1	ABSTRACT	0	1	0	1	1
CONCERN	0	1	0	1	1	CLOCK	0	1	0	1	1
SYNCHRONIZ	0	1	0	1	1	INDEXES	0	1	0	1	1
REPEATABIL	0	0	1	1	1	PRIVACY	0	0	1	1	1
COMPUTINGR	0	0	1	1	1	SUMMARIZED	0	1	0	1	1
EMPIRICAL	0	1	0	1	1	MULTI	0	1	0	1	1
ORGANIZED	0	1	0	1	1	PAGES	0	1	0	1	1
COLLECTED	0	1	0	1	1	FAULTS	0	1	0	1	1
INSTANTS	0	1	0	1	1	ASSESSING	0	1	0	1	1
ONPERFORMA	0	1	0	1	1	ALLOCATED	0	1	0	1	1
SUPERVISOR	0	0	1	1	1	ESTIMATING	1	0	0	1	1
ITH	0	1	0	1	1	JTH	0	1	0	1	1
ACCOUNTS	0	1	0	1	1	QUANTITIES	0	1	0	1	1
ARBITRARY	0	0	1	1	1	PROPORTION	0	0	1	1	1
PROBABILIT	0	0	1	1	1	CONSIDERIN	0	0	1	1	1
GHEORY	0	0	1	1	1	MARKOV	0	0	1	1	1
CHAINS.	0	0	1	1	1	OBTAINING	0	0	1	1	1
ESTIMATES	0	0	1	1	1	AVOIDING	0	0	1	1	1
NECESSITY	0	0	1	1	1	INVOLVING	0	0	1	1	1
MATRICES	0	0	1	1	1	EXPLORED	0	1	0	1	1
BUFFER	0	1	0	1	1	STORED	0	1	0	1	1

1 TOTAL NO WORDS 926
CATEGORY IS 4.20

WORD	TITL	ABST	KEY	DB FR	TOT	WORD	TITL	ABST	KEY	DB FR	TOT
PROGRAMMIN	6	30	20	21	56	LANGLAGES	4	14	17	16	35
COMPUTER	3	18	11	17	32	SOFTWARE	3	11	6	7	20
STRUCTURES	0	14	5	12	19	PROCESSING	1	6	11	12	18
DISPLAY	2	10	6	3	19	COMPILER	2	9	5	7	16
INFORMATIO	0	12	4	12	16	INDEPENDEN	0	9	5	5	14
SYNTAX	0	8	6	9	14	MULTIPROGR	3	5	6	5	14
IMPLEMENTA	2	10	2	9	14	ALGORITHM	1	12	1	8	14
TABLES	3	7	4	4	14	STORAGE	0	7	6	6	13
USER	0	12	1	6	13	NETWORK	0	10	3	5	13
DEFINITION	1	11	1	7	13	ALGORITHMS	1	10	2	4	13
GRAMMARS	1	10	1	2	12	SCHEDULING	1	8	3	4	12
REPRESENTA	1	8	3	8	12	EQUATIONS	2	9	1	2	12
SIMULATION	1	4	7	3	12	ARRAY	1	8	3	2	12
POSSIBLE	0	11	0	9	11	PROCESSES	1	8	2	6	11
IMPLEMENTE	1	10	0	10	11	METHOD	1	8	2	8	11
MACRO	0	7	3	6	10	INPUT	0	6	4	6	10
PARALLEL	1	4	5	5	10	INSTRUCTIO	0	9	0	3	9
ALGOL	1	6	2	3	9	EXTENSION	1	7	0	3	8
FORMAL	1	6	1	5	8	SEMANTICS	1	3	4	5	8
ORGANIZATI	1	5	2	5	8	DEFINED	0	8	0	7	8
NETWORKS	2	5	1	3	8	GRAPHS	1	5	2	4	8
SPECIFICAT	1	6	1	4	8	FUNCTIONS	0	8	0	5	8
COMPILERS	0	3	4	5	7	DESCRIPTIO	0	7	0	7	7
CRITERIA	1	5	1	2	7	OUTPUT	0	5	2	6	7
EXAMPLES	0	7	0	7	7	LISP	1	3	3	3	7
CONTEXT	0	6	1	4	7	INTERACTIV	1	5	1	4	7
GENERATING	3	4	0	4	7	COORDINATE	0	6	1	1	7
PARSE	1	6	0	1	7	PARSING	0	4	3	2	7
ARRAYS	1	5	1	2	7	PLOTS	0	6	1	1	7
ENTRY	1	5	1	4	7	AFFIX	1	6	0	1	7
MASK	2	2	3	1	7	PARTITIONI	1	4	2	1	7
FACILITY	1	5	0	3	6	COMPUTERS	0	4	2	5	6
APPLICATIO	0	5	1	6	6	COMPUTATIO	1	4	1	5	6
VARIABLES	0	6	0	5	6	ENVIRONMEN	2	4	0	5	6
MANIPULATI	2	1	3	4	6	DIGITAL	1	4	1	3	6
TRANSFORMA	0	6	0	3	6	SPECIFIED	0	6	0	5	6
SIMULTANEO	0	4	2	2	6	TRANSLATOR	0	3	3	4	6
GRAPH	1	2	3	2	6	DRIVEN	1	3	2	2	5
CONVERSION	3	1	2	4	6	GRASPE	0	6	0	1	5
STRUCTURED	1	4	1	3	6	OPTIMUM	0	6	0	1	6
PROCESSOR	1	1	3	4	5	CONSTRUCTI	0	5	0	4	5
LOGICAL	0	3	2	4	5	EFFICIENCY	1	2	2	4	5
EXISTING	0	5	0	4	5	REFERENCE	1	3	1	3	5
MANUAL	0	5	0	3	5	DEBUGGING	1	2	2	3	5
ASP	1	4	0	1	5	ILLUSTRATE	0	5	0	5	5
DERIVE	0	5	0	2	5	EXECUTION	0	5	0	3	5
PERMITS	0	5	0	5	5	AXIOMATIC	2	1	2	2	5

ADDRESSING	0	3	2	2	5	MODULES	0	4	1	2	5
BLOCKS	0	3	2	2	5	COMPOSITIO	1	3	1	2	5
PRESETS	0	5	0	5	5	CPU	1	3	1	1	5
FORTRAN	0	3	1	3	4	COMPUTING	1	3	0	3	4
LISTS	0	3	1	3	4	IDENTIFICA	1	2	1	1	4
STANDARDIZ	0	3	1	1	4	GOALS	0	4	0	3	4
SELF	0	2	2	2	4	CONVENIENT	0	4	0	3	4
CALCULATIO	0	4	0	2	4	SYMBOLIC	1	2	1	3	4
CURVILINEA	0	3	1	1	4	EQUATION	0	2	2	1	4
RELATIVE	0	4	0	4	4	RETRIEVAL	0	3	1	2	4
REQUESTS	0	4	0	2	4	GENERATION	0	3	1	4	4
SIMULATING	1	3	0	3	4	INTERRUPT	0	3	1	2	4
SCHEME	0	4	0	2	4	CONCURRENT	0	3	1	2	4
COMPACT	1	2	1	1	4	GARBAGE	1	2	1	1	4
BRIEFLY	0	4	0	4	4	MULTIPLE	0	3	1	3	4
STRING	0	3	1	3	4	EVALUATION	0	3	1	3	4
ISSUES	0	4	0	1	4	HARDWARE	0	2	2	3	4
RATIONAL	1	2	1	2	4	FACILITATE	0	4	0	2	4
DYNAMIC	1	2	1	2	4	NOTATION	0	3	0	3	3
EXTENSIONS	0	3	0	3	3	SOPHISTICA	0	3	0	3	3
MACROS	0	3	0	3	3	CODING	1	2	0	3	3
SYNTACTIC	0	3	0	2	3	EXPLICIT	0	3	0	3	3
VIRTUAL	0	2	1	2	3	SYNCHRONIZ	0	2	1	3	3
TESTING	0	2	1	3	3	ASSOCIATIV	1	1	1	1	3
ARBITRARY	0	3	0	3	3	OUTLINED	0	3	0	2	3
MODELING	1	1	1	2	3	TRAC	0	2	1	1	3
TRULY	1	2	0	2	3	DERIVED	0	3	0	2	3
60	0	3	0	2	3	HANDLED	0	3	0	3	3
GENERALITY	0	3	0	3	3	ASSEMBLER	0	2	1	2	3
FINITE	0	1	2	2	3	ALGORITHMI	0	1	2	3	3
OPTIMIZATI	0	3	0	2	3	DERIVING	1	2	0	3	3
MATHEMATIC	0	2	1	2	3	DEMONSTRAT	0	3	0	3	3
NAVIER	0	2	1	1	3	CONTINUITY	0	2	1	1	3
ORIENTED	0	2	1	2	3	REQUEST	1	2	0	2	3
LOGIC	0	3	0	1	3	HANDLING	0	2	1	3	3
STRUCTURAL	0	2	1	2	3	SIMULATED	0	3	0	2	3
TESTED	0	3	0	3	3	PLEX	0	1	2	1	3
CASES	0	3	0	2	3	ADVANTAGES	0	3	0	3	3
PROVING	0	2	1	2	3	INDUCTION	1	0	2	2	3
APAREL	1	2	0	1	3	CODE	0	3	0	2	3
ALTERNATIV	0	3	0	3	3	SEMANTIC	1	2	0	2	3
TRANSLATIO	1	1	1	1	3	PUSHDOWN-	0	2	1	1	3
CONSISTS	0	3	0	3	3	MACHINES	0	2	1	3	3
CONCEPTS	0	2	1	3	3	ACCESS	0	3	0	2	3
EFFICIENT	0	3	0	3	3	SHARING	0	2	1	1	3
COMMUNICAT	0	2	1	3	3	PLOT	0	2	1	1	3
MAP	0	2	1	2	3	SUBPLOTS	0	3	0	1	3
GRAPHICS	0	1	2	2	3	APPLICABLE	0	3	0	3	3
FLOWCHART	0	2	1	2	3	FILES	0	2	1	1	3
DISCUSSES	0	3	0	3	3	68	1	2	0	1	3
RECURSIVE	0	2	1	2	3	DEVELOPMEN	0	2	1	3	3
ADVANCED	1	1	1	1	3	KOSTER	0	3	0	1	3
EMULATION	1	1	1	1	3	OPTIMAL	1	1	1	1	3
APPROACHIN	0	3	0	1	3	PORTABLE	1	1	1	1	3
TRACE	1	1	1	1	3	FUNCTIONAL	0	2	0	2	2

ALGEBRAIC	0	2	0	2	2	ASSEMBLY	0	2	0	2	2
EMBEDDING	0	2	0	2	2	FOUNDATION	0	2	0	2	2
ASYNCHRONO	0	2	0	2	2	REPRODUCIB	0	1	1	1	2
CONTENT	0	2	0	2	2	NOTION	0	2	0	1	2
CAPABILITY	0	2	0	2	2	DETERMINIS	0	0	2	1	2
RING	1	0	1	1	2	RELATIONSH	0	2	0	2	2
MANIPULATE	0	2	0	2	2	PAGED	1	0	1	1	2
STORED	0	2	0	2	2	DETAILS	0	2	0	2	2
PAGING	0	1	1	2	2	SECONDARY	0	1	1	2	2
ACCOMMODAT	1	1	0	1	2	RELIABLE	0	2	0	1	2
ACHIEVING	0	2	0	2	2	64	0	1	1	1	2
VARIATIONS	0	2	0	2	2	TRADEMARK	0	1	1	1	2
GPL	1	1	0	1	2	DECLARATIO	0	2	0	2	2
POINTERS	0	2	0	2	2	EXTENDING	0	1	1	2	2
GENERATOR	0	1	1	2	2	NETWORK	0	2	0	1	2
NEWROK	0	2	0	1	2	MIXTURE	0	2	0	2	2
INTERPRETI	0	2	0	2	2	INCREMENTA	0	1	1	1	2
LRLTRAN	1	1	0	1	2	MINIMIZES	0	2	0	2	2
POLISH	0	1	1	1	2	DEPENDENT	0	2	0	2	2
PHYSICS	1	1	0	1	2	STOKES	0	1	1	1	2
ORTHOGONAL	0	2	0	2	2	THEORETICA	0	2	0	2	2
CONSIDERAT	0	2	0	2	2	POLAR	0	2	0	1	2
TENSOR	0	0	2	1	2	CODAS	1	1	0	1	2
DISPLAYED	0	2	0	2	2	FORMAT	0	2	0	2	2
EXCLUSIVE	1	1	0	1	2	EASE	0	2	0	2	2
REDUCTION	0	2	0	2	2	TASKS	0	2	0	2	2
CONVERTING	0	2	0	2	2	PRECISE	0	2	0	2	2
AUTHOR	0	2	0	2	2	FEELS	0	2	0	2	2
FEATURE	0	2	0	2	2	ERRORS	0	2	0	2	2
CONTRIBUTE	0	2	0	2	2	PRIME	0	2	0	2	2
ROUTINES	0	2	0	2	2	PL360	0	1	1	1	2
SEQUENTIAL	0	2	0	2	2	CHAINED	0	2	0	1	2
COMBINATIO	0	2	0	2	2	COLLECTION	0	1	1	1	2
PRIMITIVE	0	1	1	2	2	OCCUR	1	1	0	1	2
CONVENTION	0	2	0	2	2	HANDLE	0	2	0	2	2
PRIORITY	0	2	0	2	2	COMMONLY	0	2	0	2	2
ATTEMPT	0	2	0	2	2	INVOLVES	0	2	0	2	2
AXIOMS	0	2	0	1	2	PROOF	0	1	1	1	2
ARGUED	0	2	0	2	2	TOPICS	0	1	1	2	2
OPERATOR	0	0	2	2	2	MATCHING	0	1	1	1	2
BNF	0	1	1	1	2	EXECUTED	0	2	0	2	2
PREPROCESS	0	1	1	2	2	PARSER	0	1	1	1	2
TEXT	0	1	1	2	2	TRANSDUCTI	0	1	1	1	2
DESCRIBE	0	2	0	2	2	AUTOMATA	0	1	1	1	2
FRAMEWORK	0	2	0	2	2	SIGNIFI	0	1	1	2	2
MACHINE;	0	2	0	1	2	MANUFACTUR	0	2	0	2	2
DIAGNOSTIC	0	2	0	2	2	AIDS	0	2	0	2	2
CONSTRUCTE	0	2	0	2	2	GENERALIZE	1	1	0	2	2
VECTORS	0	2	0	2	2	DIMENSIONA	0	2	0	1	2
MULTIDIMEN	0	1	1	1	2	CODEWORDS	0	2	0	1	2
CONTAINING	0	2	0	2	2	MATRIX	0	1	1	2	2
GEDANKEN	1	1	0	1	2	TYPELESS	1	0	1	1	2
PERMITTED	0	2	0	1	2	PERMISSIBL	0	2	0	1	2
LABELS	0	2	0	2	2	REFERENCES	0	2	0	2	2
EQUIVALENT	0	2	0	2	2	QUASI	0	1	1	1	2

TREATING	1	1	0	2	2	COMPARISON	0	2	0	2	2
TRAFFIC	0	1	1	1	2	CONTROLLED	0	2	0	2	2
GENERATE	0	2	0	2	2	COMMANDS	0	2	0	1	2
ACCOMPLISH	0	2	0	2	2	EMBEDDED	0	2	0	2	2
INTERACTIO	0	0	2	2	2	CONDITIONA	0	1	1	2	2
COMMENT	2	0	0	2	2	PLOTTING	1	0	1	1	2
ENABLES	0	2	0	2	2	BRIEF	0	2	0	2	2
SCALED	0	1	1	1	2	FORMULATED	0	2	0	2	2
SCALING	0	2	0	1	2	INTERACT	0	2	0	2	2
ADDRESSES	1	1	0	1	2	COMPILING	0	1	1	2	2
MUTUAL	0	1	1	2	2	COVERING	0	1	1	1	2
INTEGER	0	1	1	2	2	RECURSION	1	0	1	1	2
HOST	0	2	0	2	2	ALTRAN	1	1	0	1	2
CONSIDERAB	0	2	0	2	2	PERMITTING	0	2	0	2	2
SUGGESTS	0	2	0	2	2	PRIMITIVES	0	1	1	2	2
BLISS	1	1	0	1	2	SPECIFIC	0	1	1	2	2
SHARED	0	1	1	1	2	REGIONS	0	1	1	1	2
PROGRAMMER	0	2	0	2	2	DESIRED	0	2	0	2	2
FUTURE	1	1	0	1	2	REQUIREMEN	1	1	0	1	2
STORES	0	2	0	1	2	INVOCATION	0	2	0	1	2
VAN	0	2	0	1	2	WIJNGAARDE	0	2	0	1	2
FLOYD	0	1	1	1	2	MICROPROGR	0	1	1	1	2
CONDITIONS	0	2	0	2	2	FLOWCHARTS	1	1	0	1	2
FINDING	0	2	0	1	2	SUBARRAYS	0	2	0	1	2
PERFORMANC	0	1	1	1	2	IMPLEMENTI	0	1	0	1	1
SATISFACTO	0	1	0	1	1	CONTENDED	0	1	0	1	1
CONSTITUES	0	1	0	1	1	CANONICAL	0	1	0	1	1
SUPPORTING	0	1	0	1	1	CONFLICT	0	1	0	1	1
UNIVERSAL	0	1	0	1	1	METALANGUA	0	1	0	1	1
METAPHYSIC	0	1	0	1	1	ALLOCATION	0	1	0	1	1
FALSE	0	1	0	1	1	ECONOMICS	0	1	0	1	1
FORTRANAUT	0	1	0	1	1	SURVEY	0	1	0	1	1
DESIGNING	1	0	0	1	1	DESIGNER	0	1	0	1	1
ADOPT	0	1	0	1	1	ACCEPTING	0	1	0	1	1
REJECTING	0	1	0	1	1	CRITERIAOF	0	1	0	1	1
RECORDABIL	0	1	0	1	1	SPECIFIABI	0	1	0	1	1
IMPLY	0	1	0	1	1	DESIRES	0	1	0	1	1
INFLUENCES	0	1	0	1	1	AFFECTING	0	1	0	1	1
DEFINE	0	1	0	1	1	SCOPE	0	1	0	1	1
ABSTRACT	0	1	0	1	1	CONCERN	0	1	0	1	1
CLOCK	0	1	0	1	1	INDEXES	0	1	0	1	1
IPERATING	0	0	1	1	1	REPEATABIL	0	0	1	1	1
PRIVACY	0	0	1	1	1	COMPUTINGR	0	0	1	1	1
PACKAGE	0	1	0	1	1	ANARBITRAR	0	1	0	1	1
DRAWING	0	1	0	1	1	APPENDIX	0	1	0	1	1
ENABLE	0	1	0	1	1	READER	0	1	0	1	1
EXPLODED	0	1	0	1	1	CORE	0	1	0	1	1
BUFFER	0	1	0	1	1	DRUM	0	1	0	1	1
TIMING	0	1	0	1	1	UTILIZATIO	0	0	1	1	1
ADOPTED	0	1	0	1	1	OPPRESSIVE	0	1	0	1	1
RIGIDITY	0	1	0	1	1	AVOIDED	0	1	0	1	1
ACCESSIBLE	0	1	0	1	1	UNDERTAKEN	0	1	0	1	1
RELYING	0	1	0	1	1	USEFLLLY	0	1	0	1	1
PERFORMED	0	1	0	1	1	MINIMAL	0	1	0	1	1
ELEMENT	0	1	0	1	1	CONSTRUCTS	0	1	0	1	1

UNDERLINED	0	1	0	1	1	IMPLICITLY	0	1	0	1	1
USAGE	0	1	0	1	1	DEFINABLE	0	1	0	1	1
PLACED	0	1	0	1	1	ANALYZING	1	0	0	1	1
[redacted]	1	0	0	1	1	PREPARATIO	0	1	0	1	1
DEFINING	0	1	0	1	1	PERFORMING	0	1	0	1	1
MANAGEMENT	0	1	0	1	1	AUTOPERT	0	0	1	1	1
REVISED	0	0	1	1	1	REMOTE	0	1	0	1	1
TERMINAL	0	1	0	1	1	BURROUGHS	0	1	0	1	1
B5500	0	1	0	1	1	COMBINES	0	1	0	1	1
COMPILATIO	0	0	1	1	1	EXTENSIVE	0	1	0	1	1
CONFRONT	0	1	0	1	1	POSSESSES	0	1	0	1	1
ACQUIRES	0	1	0	1	1	COHESION	0	1	0	1	1
DEPENDENCI	0	1	0	1	1	ISOLATES	0	1	0	1	1
AUGMENTED	0	1	0	1	1	INTERNALLY	0	1	0	1	1
POSTFIX	0	1	0	1	1	REFERRED	0	1	0	1	1
COMPOSITE	0	1	0	1	1	BLOCKING	0	1	0	1	1
OCCURS	0	1	0	1	1	LOOP	0	1	0	1	1
SCATTER	0	0	1	1	1	SUBSEGMENT	0	0	1	1	1
WHEREBY	0	1	0	1	1	REQUESTED	0	1	0	1	1
EFFECTIVEN	0	1	0	1	1	FLUID	0	1	0	1	1
MOTION	0	1	0	1	1	RELATING	0	1	0	1	1
CARTESIAN	0	1	0	1	1	TRIAD	0	1	0	1	1
SUPPLIED	0	1	0	1	1	METHODOLOG	0	1	0	1	1
CYLINDRICA	0	1	0	1	1	SPHERICAL	0	1	0	1	1
FORMAC	0	0	1	1	1	CUSTOMER	0	1	0	1	1
SPECIFYING	0	1	0	1	1	PUBLICATIO	0	1	0	1	1
GRAPHIC	0	0	1	1	1	DETAILED	0	1	0	1	1
OBJECTIVES	0	1	0	1	1	SOUGHT	0	1	0	1	1
IMPROVED	0	1	0	1	1	SEPARATION	0	1	0	1	1
BEHAVIORAL	0	1	0	1	1	BOOLEAN	0	1	0	1	1
ELIMINATIO	0	1	0	1	1	DETERIORAT	0	1	0	1	1
QUEUE	0	1	0	1	1	GROWS	0	1	0	1	1
DEALS	0	1	0	1	1	EFFECTIVEL	0	1	0	1	1
SERIAL	0	1	0	1	1	PASSAGE	0	1	0	1	1
QUANTITATI	0	1	0	1	1	COMBINATIO	0	1	0	1	1
SYNCHRONOU	0	1	0	1	1	QUEUING	0	0	1	1	1
POTENT	0	1	0	1	1	PITFALLS	0	1	0	1	1
UNRECLAIL	0	1	0	1	1	PSEUDOCONC	0	1	0	1	1
SUPPOSEDLY	0	1	0	1	1	INTENDED	0	1	0	1	1
INCLUDES	0	1	0	1	1	TERMINATIO	0	1	0	1	1
REPRESENTI	0	1	0	1	1	THEREBY	0	1	0	1	1
HYPOTHETIC	0	1	0	1	1	IBM	0	1	0	1	1
360	0	1	0	1	1	PARALLELIS	0	0	1	1	1
COLLECTIO	1	0	0	1	1	OPERATORS	0	1	0	1	1
COLLECTOR	0	1	0	1	1	ATTEMPTS	0	1	0	1	1
COMPACTI	0	1	0	1	1	RELOCATES.	0	1	0	1	1
REARRANGES	0	1	0	1	1	TEMPORARY	0	1	0	1	1
COMPACTING	0	1	0	1	1	IMPLEMENT	0	1	0	1	1
RELOCATION	0	0	1	1	1	RECLAMATIO	0	0	1	1	1
POINTER	0	0	1	1	1	SIMULATIG	0	1	0	1	1
DISCRETE	0	1	0	1	1	DEALT	0	1	0	1	1
INTERCONNE	0	1	0	1	1	IGNORED	0	1	0	1	1
DEALING	0	1	0	1	1	INTRODUCED	0	1	0	1	1
ABSTRACTIO	0	1	0	1	1	ADMITS	0	1	0	1	1
MAJORITY	0	1	0	1	1	ENCOUTENRE	0	1	0	1	1

1 TOTAL NO WORDS 950
CATEGORY IS 4.30

WORD	TITL	ABST	KEY	DB FR	TOT	WORD	TITL	ABST	KEY	DB FR	TOT
STORAGE	6	24	21	11	51	COMPUTER	3	32	14	26	49
SHARING	5	20	23	23	48	MULTIPROGR	4	12	24	21	40
PROCESSES	2	23	7	11	32	PAGING	2	7	20	15	29
SCHEDULING	1	16	11	11	28	INFORMATIO	2	18	7	9	27
PERFORMANC	3	12	11	10	26	PROGRAMMIN	3	10	11	11	24
PROCESSING	3	9	12	12	24	USER	1	19	4	12	24
IMPLEMENTA	3	17	1	14	21	COMMUNICAT	4	12	5	12	21
ALLOCATION	2	7	11	8	20	VIRTUAL	2	7	11	11	20
SOFTWARE	1	14	5	9	20	ALGORITHM	1	15	4	9	20
MEASUREMEN	1	6	8	5	15	HARDWARE	0	14	0	10	14
SEGMENTATI	0	5	8	7	13	ALGORITHMS	0	5	8	9	13
ACCESS	1	6	6	7	13	INPUT	0	7	6	7	13
SUPERVISOR	1	10	2	4	13	DISPLAY	2	6	3	6	13
ENVIRONMEN	3	7	2	10	12	CRITERIA	2	8	2	5	12
AUTOMATIC	2	4	4	4	11	APPLICATIO	1	9	1	9	11
RESOURCE	2	5	4	6	11	SIMULATION	1	5	5	4	11
HIERARCHY	1	5	5	4	11	EXECUTION	0	11	0	9	11
COMPUTING	3	4	4	4	11	GRAPHIC	0	8	3	3	11
OVERLAY	1	6	2	2	10	INDEPENDEN	0	10	0	8	10
OUTPUT	0	6	4	7	10	PARALLEL	0	5	5	4	10
DYNAMIC	1	4	4	6	9	CHARACTERI	1	8	0	7	9
MODELS	1	2	6	7	9	INSTRUCTIO	0	6	3	4	9
MATRIX	1	7	1	2	9	FOLDING	1	5	3	2	9
EDITING	1	4	4	2	9	FACILITY	1	7	0	3	8
MULTICS	2	3	3	3	8	MULTIPROCE	1	1	6	7	8
EXTENDED	1	5	2	5	8	FUNCTIONS	0	7	1	6	8
SHARED	2	4	2	5	8	TERMINAL	0	8	0	6	8
SEMIAUTOMA	0	2	4	2	7	REPRESENTA	0	5	2	3	7
COLLECTION	0	7	0	5	7	PAGED	3	3	1	5	7
ORIENTED	0	5	2	4	7	EXISTING	0	7	0	5	7
UTILIZATIO	1	4	2	3	7	LANGUAGES	0	5	2	5	7
ENTRY	1	5	1	3	7	BATCH	1	4	1	3	6
IMPLEMENTE	0	6	0	6	6	PAGES	0	6	0	4	6
DEFINED	0	6	0	5	6	PRIORITY	1	3	2	3	6
MECHANISM	0	6	0	4	6	CONSOLE	1	5	0	2	6
PERFORM	0	6	0	5	6	REMOTE	1	2	3	3	6
USERS	0	6	0	5	6	PARAMETERS	0	6	0	2	6
INTERACTIV	3	3	0	3	6	INTERACTIO	0	3	3	5	6
INTERFEREN	1	4	1	3	6	TOOLS	0	5	1	2	6
IBM	0	2	3	3	5	PROCESSOR	1	1	3	4	5
360	0	3	2	3	5	UTILITY	1	3	1	3	5
COPE	1	3	1	2	5	CONFIGURAT	0	5	0	3	5
SYNCHRONIZ	1	3	1	3	5	MULTI	0	4	1	5	5
BUFFER	0	5	0	2	5	CONTEXT	0	4	1	2	5
SCHEME	1	4	0	3	5	EXERCISES	1	3	1	1	5
GRAPHICS	0	1	4	2	5	COMMANDS	0	5	0	3	5
TEXT	0	3	2	2	5	CIRCUIT	1	3	1	1	5

8600	0	3	1	3	4	FORTRAN	0	3	1	3	4
CAPABILITI	0	4	0	3	4	ADDRESSING	0	2	2	3	4
DEVELOPMEN	0	4	0	3	4	COMPUTATIO	0	2	2	3	4
CONCEPTS	0	3	1	4	4	COMPUTERS	0	1	3	3	4
ABSTRACTIO	0	3	1	3	4	CORRECTNES	0	3	1	3	4
DEBUGGING	1	0	3	2	4	ANALYZED	0	4	0	3	4
POSSIBLE	0	4	0	4	4	DISTRIBUTE	0	2	2	3	4
EVALUATION	0	3	1	3	4	REPLACEMEN	0	2	2	3	4
ESTIMATION	0	2	2	1	4	PROGRAMMER	0	4	0	3	4
STATISTICA	1	2	1	2	4	JOB	0	4	0	3	4
EXAMINED	0	4	0	4	4	MANIPULATI	0	4	0	3	4
SPECIFICAT	0	4	0	4	4	INTERFACE	0	3	1	3	4
MANUAL	1	3	0	2	4	MONITORING	1	2	1	1	4
TAPE	0	2	2	2	4	OVERHEAD	0	3	1	3	4
EFFICIENT	1	3	0	4	4	CONVERSATI	1	1	2	3	4
DESCRIPTIO	0	4	0	4	4	LINC	0	3	1	1	4
FILING	0	2	2	1	4	MEASURING	0	3	1	2	4
AIDED	1	2	1	1	4	HASHING	1	2	1	1	4
NETWORK	1	2	1	2	4	PROVE	0	4	0	2	4
PROPOSAL	1	3	0	2	4	OVERLAYS	1	3	0	1	4
7090	0	0	3	1	3	7094	0	0	3	1	3
SYSTEMS	0	1	2	2	3	COMPILER	0	1	2	3	3
LINKAGE	0	1	2	3	3	METHOD	1	2	0	2	3
HIERARCHIE	0	0	3	2	3	MEDIUM	1	2	0	2	3
BALANCING	0	3	0	2	3	COMBINED	0	3	0	2	3
QUANTITIES	0	3	0	3	3	ESTIMATING	1	2	0	3	3
COMPARED	0	3	0	3	3	DEVICES	0	2	1	3	3
DIVIDED	0	3	0	2	3	SEQUENTIAL	0	2	1	1	3
LOGICAL	0	3	0	3	3	PRIMITIVES	0	2	1	3	3
COOPERATIN	0	1	2	3	3	DESIGNER	0	3	0	2	3
EXAMPLES	0	3	0	3	3	CAPABILITY	0	3	0	3	3
PRIVACY	0	1	2	2	3	CONTROLLED	0	2	1	3	3
ACCOMPLISH	0	3	0	3	3	FAULTS	0	3	0	2	3
CASES	0	3	0	3	3	DERIVED	0	2	1	3	3
MATRICES	1	1	1	3	3	IMPROVEMEN	0	2	1	3	3
EFFICIENT	0	3	0	2	3	EXECUTE	0	3	0	3	3
OVERALL	0	3	0	3	3	REQUIREMEN	0	3	0	3	3
PROCESSORS	0	2	1	3	3	MATHEMATIC	0	1	2	3	3
STORED	0	2	1	2	3	ORGANIZATI	1	0	2	2	3
EFFICIENCY	0	2	1	3	3	ROW	0	3	0	2	3
ARRAY	0	1	2	2	3	INTERFACES	0	2	1	2	3
SEQUENCES	0	3	0	2	3	REPRESENTE	0	3	0	2	3
YIELDS	0	3	0	2	3	GRADING	1	1	1	1	3
OPERATORS	0	2	1	2	3	DECREASE	0	3	0	1	3
MACHINES	0	0	3	3	3	GRAPHICAL	1	1	1	1	3
HUMAN	0	3	0	1	3	INSTRUMENT	1	0	2	2	3
INTENDED	0	3	0	3	3	SEGMENTED	0	3	0	2	3
BIOMEDICAL	0	1	2	1	3	STRUCTURIN	0	3	0	2	3
LAP6	0	2	1	1	3	SCROLL	0	2	1	1	3
FILED	0	3	0	1	3	DIRECTORY	0	2	1	2	3
CONCURRENT	1	2	0	3	3	READERS	1	2	0	1	3
SOLUTIONS	0	3	0	2	3	GOALS	0	3	0	1	3
ARCHITECTU	0	2	1	1	3	MICROPROGR	0	2	1	1	3
NORMALLY	0	3	0	2	3	CONTROLLAB	0	3	0	1	3

LOADER	0	0	2	2	2	FUNCTIONAL	0	1	1	2	2
MEMORIES	0	1	1	2	2	MECHANISMS	0	0	2	2	2
TRACING	0	1	1	2	2	RELATIVE	0	2	0	1	2
VARIATIONS	0	2	0	2	2	VITAL	0	2	0	2	2
FORMULATED	0	2	0	2	2	DEMANDS	0	2	0	2	2
CONSIDERAT	1	1	0	2	2	MULTIPLE	1	0	1	1	2
ADDRESSABL	0	2	0	2	2	EXECUTABLE	0	2	0	2	2
RANDOM	0	2	0	1	2	ARRANGEMEN	0	2	0	2	2
CONVENTION	0	2	0	2	2	SLOWER	0	2	0	2	2
VERIFICATI	0	1	1	1	2	BUFFERING	0	0	2	2	2
FRAMEWORK	0	2	0	2	2	VECTOR	0	2	0	2	2
RELATIONSH	0	2	0	2	2	INTERRUPT	0	1	1	1	2
ESTABLISHE	0	2	0	2	2	DESIGNING	1	1	0	2	2
REPRODUCIB	0	1	1	1	2	CONTENT	0	2	0	2	2
DEFINE	0	2	0	2	2	SCOPE	0	2	0	2	2
NOTION	0	2	0	1	2	DETERMINIS	0	0	2	1	2
TESTING	0	1	1	2	2	FLEXIBLE	0	2	0	2	2
RELIABLE	0	1	1	2	2	ENTERPRISE	0	1	1	1	2
DEPENDENT	0	2	0	2	2	MULTIACCES	0	2	0	2	2
INSTALLATI	0	1	1	1	2	CONSTRUCTI	0	2	0	2	2
OFC	2	0	0	2	2	ORGANIZED	0	2	0	2	2
INTERPRETI	0	2	0	2	2	DESCRIBE	0	2	0	2	2
FREQUENCY	0	2	0	1	2	EXECUTING	0	2	0	1	2
ALLOCATED	0	2	0	2	2	JUMPS	1	1	0	1	2
EASIER	0	1	1	1	2	ESTIMATE	0	1	1	1	2
MARKOV	0	1	1	2	2	PARALYSIS	1	1	0	1	2
FUTURE	0	2	0	2	2	BRIEFLY	0	2	0	2	2
DIFFICULT	0	2	0	2	2	CONSOLES	0	2	0	1	2
COMBINES	0	2	0	2	2	INCREMENTA	0	1	1	1	2
COMPILERS	0	0	2	2	2	ADEPT	0	2	0	1	2
CATALOGED	0	2	0	1	2	FLEXIBILIT	0	2	0	2	2
ASSISTED	0	2	0	2	2	RICE	0	2	0	1	2
LABELED	0	2	0	1	2	CODEWORDS	0	1	1	1	2
BLOCKS	0	2	0	1	2	ADVANTAGES	0	2	0	2	2
PROGRAMMED	0	2	0	2	2	STRUCTURES	0	1	1	1	2
EXPERIENCE	0	2	0	2	2	ALLOWS	0	2	0	2	2
SUBOPERATI	0	1	1	1	2	VERSION	0	2	0	2	2
FEATURE	0	2	0	2	2	SUBMONITOR	0	1	1	2	2
UTILIZE	0	2	0	1	2	POTENTIALL	0	2	0	2	2
EMBEDDED	0	2	0	2	2	REFERENCES	0	2	0	2	2
TRANSFORMA	0	2	0	2	2	INITIAL	0	2	0	2	2
ASSEMBLY	0	2	0	2	2	PROVISION	0	2	0	2	2
SUPPLIED	0	2	0	1	2	QUANTUM	0	2	0	2	2
GRAPHICALL	0	2	0	1	2	CALCULATIO	0	2	0	2	2
MARKOVIAN	0	0	2	2	2	ACCOMPANIE	0	2	0	1	2
INFORMAL	0	2	0	2	2	FORMAL	0	2	0	2	2
RESIDENT	0	2	0	1	2	AUXILIARY	0	2	0	2	2
COMPREHENS	0	2	0	2	2	DCF	0	2	0	1	2
MTS	0	2	0	1	2	OUTLINED	0	2	0	2	2
IDENTIFIED	0	2	0	1	2	RECORDED	0	2	0	1	2
OCCURRENCE	0	2	0	2	2	NELSON	0	2	0	1	2
MINIMIZE	0	2	0	2	2	TASKS	0	2	0	2	2
HANDLING	0	2	0	2	2	NUCLEUS	1	1	0	1	2

MANIPULATE	0	2	0	2	2	LABORATORY	0	2	0	2	2
MIT	0	2	0	2	2	GENERATE	0	2	0	2	2
KEYBOARD	0	2	0	2	2	LIMITATION	0	2	0	2	2
PERTINENT	0	2	0	2	2	PSEUDO	1	0	1	2	2
2048	1	1	0	1	2	PREPARATIO	0	2	0	1	2
UNIFORM	0	2	0	2	2	AUGMENTED	0	2	0	2	2
TAPES	0	2	0	1	2	AID	0	2	0	2	2
EXTENSION	0	2	0	2	2	ENTRIES	0	2	0	1	2
SIMILARITY	0	2	0	1	2	WHEREAS	0	2	0	2	2
REGARDED	0	2	0	2	2	NAMELIST	0	2	0	1	2
VALUELIST	0	2	0	1	2	VIEWED	0	2	0	2	2
TX	0	1	1	1	2	SIMULTANEO	0	2	0	2	2
DISTINCT	0	2	0	2	2	DELAY	0	2	0	2	2
CATHODE	0	1	1	1	2	RAY	0	1	1	1	2
TUBE	0	1	1	1	2	ACHIEVE	0	2	0	1	2
CHECKOUT	0	2	0	2	2	VENUS	1	1	0	1	2
COMBINATIO	0	2	0	2	2	SEGMENTS	0	1	1	2	2
DEADLOCK	0	1	1	2	2	HANDLES	0	2	0	1	2
ABLE	1	1	0	2	2	MESSAGES	0	2	0	1	2
OPERATE	0	2	0	2	2	MODELING	0	0	2	2	2
DEFINITION	0	2	0	1	2	ASSUMPTION	0	2	0	2	2
TRANSLATIO	0	2	0	1	2	CONTEMPORA	0	2	0	1	2
VIA	1	1	0	2	2	FILES	0	2	0	2	2
MULTIPLEXE	0	1	1	1	2	REACTIONS	0	1	1	1	2
COMPARISON	0	2	0	1	2	WRITABLE	0	2	0	1	2
SUBPROBLEM	0	2	0	1	2	PARTIAL	0	2	0	1	2
CLASSIFIED	0	1	0	1	1	SUBPROGRAM	0	0	1	1	1
INTEGRATED	0	0	1	1	1	ALGOL	0	0	1	1	1
OS	0	0	1	1	1	6000	0	0	1	1	1
COMPUTERSY	0	0	1	1	1	3400	0	0	1	1	1
3600	0	0	1	1	1	INTRODUCTI	0	0	1	1	1
OVERVIEW	0	0	1	1	1	DESIGNS	0	1	0	1	1
EASING	0	1	0	1	1	UNDERLYING	0	1	0	1	1
PERSPECTIV	0	1	0	1	1	ASSESSED	0	1	0	1	1
BRIEF	0	1	0	1	1	SURVEY	0	1	0	1	1
APPENDIX	0	1	0	1	1	FRAGMENTAT	0	0	1	1	1
MEMORYPAGI	0	1	0	1	1	SYNOPSIS	0	1	0	1	1
COMPARATIV	0	1	0	1	1	PREDECESSO	0	1	0	1	1
VERSIONS	0	1	0	1	1	ALTERNATE	0	1	0	1	1
MIX	0	1	0	1	1	ABSENCE	0	1	0	1	1
MANIFESTAT	0	1	0	1	1	ONGOING	0	1	0	1	1
ACTIVITY;	0	1	0	1	1	DEFINED;	0	1	0	1	1
HEIGHTEN	0	1	0	1	1	COMPONENTS	0	1	0	1	1
DUAL	0	1	0	1	1	GENERALIZA	0	1	0	1	1
GAINS	0	1	0	1	1	TRADITIONA	0	1	0	1	1
OBSERBATIO	0	1	0	1	1	FASTER	0	1	0	1	1
CORPORATIO	0	0	1	1	1	ACCTIVITIE	0	1	0	1	1
HIERACHICA	0	1	0	1	1	SOUNDNESS	0	1	0	1	1
EVOLVED	0	1	0	1	1	SUITABLE	0	1	0	1	1
NOTIONS	0	1	0	1	1	OFPROCESS	0	1	0	1	1
FEASIBILIT	0	1	0	1	1	EMBODYING	0	1	0	1	1
INTERLOCKI	0	1	0	1	1	INTERLOCKS	0	0	1	1	1
ADOPT	0	1	0	1	1	EXPLICIT	0	1	0	1	1
ACCEPTING	0	1	0	1	1	REJECTING	0	1	0	1	1

1 TOTAL NO WORDS 952
CATEGORY IS 4.40

WORD	TITL	ABST	KEY	DB FR	TOT	WORD	TITL	ABST	KEY	DB FR	TOT
COMPUTER	4	21	22	22	47	DISPLAY	4	13	16	7	33
STORAGE	2	18	9	9	29	ALGORITHM	4	13	8	11	25
INPUT	2	13	8	10	23	USER	0	18	1	8	19
OUTPUT	1	10	7	11	18	SORTING	3	11	4	6	18
PROGRAMMIN	2	7	5	9	14	TABLES	2	8	4	3	14
PROCESSING	1	5	7	8	13	COMMUNICAT	0	6	7	8	13
DEBUGGING	2	6	4	3	12	RANDOM	2	3	7	3	12
SOFTWARE	1	6	5	4	12	VIRTUAL	1	7	3	4	11
MULTIPROGR	2	2	7	5	11	INFORMATIO	0	8	3	7	11
MONITOR	0	7	3	4	10	COLLECTION	0	7	3	6	10
GRAPHICAL	2	7	1	4	10	ALLOCATION	1	6	3	5	10
PERFORMANC	1	3	6	4	10	MEASUREMEN	0	5	5	3	10
GRAPHICS	1	1	7	4	9	REPRESENTA	1	3	5	4	9
ENVIRONMEN	3	4	2	5	9	DOCUMENTAT	1	4	4	1	9
CRITERIA	1	5	1	3	8	COMPUTING	2	3	2	4	8
COMPUTERS	2	4	2	4	8	CORRECTION	1	3	4	3	8
TERMINAL	1	3	4	5	8	BRAILLE	1	1	6	1	8
TEXT	0	7	1	4	8	IBM	1	6	1	6	8
ORGANIZATI	2	2	4	5	8	ENTRY	2	5	1	3	8
PAGING	1	4	3	3	8	SYNTAX	1	3	4	3	8
MERGING	1	2	5	3	8	CONSOLE	1	5	1	2	7
INDEPENDEN	0	5	2	2	7	GRAPHIC	0	4	3	2	7
METHOD	0	6	1	5	7	IMPLEMENTA	1	6	0	6	7
ERRORS	0	7	0	5	7	INSTRUCTIO	0	5	2	2	7
COMPUTATIO	0	5	2	4	7	STATISTICA	1	1	5	3	7
CONVERSION	1	3	3	4	7	LISP	1	4	2	4	7
IMPLEMENTE	0	7	0	7	7	360	1	5	1	6	7
PLOTS	0	6	1	1	7	MERGE	1	5	1	2	7
EXAMPLES	0	6	0	6	6	PARALLEL	0	3	3	3	6
PROCESSES	0	5	1	4	6	DUPLEX	2	2	2	1	6
SCHEME	0	6	0	3	6	RASTER	0	1	5	3	6
PROGRAMMER	0	6	0	5	6	CONVENTION	0	6	0	5	6
VARIABLES	0	6	0	3	6	CHARACTERI	0	6	0	5	6
BUFFER	1	5	0	2	6	STRUCTURES	1	4	1	3	6
POLYNOMIAL	1	4	1	1	6	DETERMINAN	1	3	2	1	6
ALGORITHMS	0	5	1	4	6	PL360	1	3	2	2	6
COMPACT	1	3	2	2	6	HARDWARE	0	5	1	4	6
SPELLING	1	4	1	1	6	EDITING	0	3	3	1	6
TV	0	1	4	1	5	DISPLAYED	0	5	0	4	5
DESCRIPTIO	0	5	0	5	5	EFFICIENT	0	5	0	5	5
OBTAINING	1	3	1	3	5	REQUIREMEN	0	5	0	5	5
CHARACTERS	0	4	1	2	5	BEZOUT	1	3	1	1	5
GENERATOR	2	3	0	2	5	GENERATORS	0	2	3	2	5
MODULES	0	5	0	2	5	SLIP	1	3	1	1	5
COLLECTOR	1	3	1	4	5	AUTOMATIC	0	4	1	3	5
TOOLS	0	4	1	1	5	DISTRIBUTI	1	4	0	2	5
SCANNED	1	3	1	1	5	POSSIBLE	0	4	0	3	4

INTERACTIO	0	2	2	4	4	GENERATION	0	4	0	4	4
DIGITAL	0	3	1	2	4	EXPERIENCE	1	3	0	3	4
ROUTINES	0	4	0	3	4	CORRECT	1	1	2	2	4
UTILITY	0	3	1	3	4	CORE	0	4	0	3	4
ESTIMATION	0	2	2	1	4	PROBABILIT	1	2	1	4	4
MULTIPLE	0	3	1	2	4	PERMITS	0	4	0	3	4
MODELS	0	2	2	3	4	SPECIFICAT	0	4	0	4	4
EXTENDED	0	4	0	3	4	EFFICIENCY	1	2	1	4	4
SHARING	1	2	1	2	4	ELIMINATIO	1	1	2	2	4
EXTREMELY	1	3	0	1	4	PORTABLE	1	3	0	1	4
IMAGES	1	1	2	2	4	MICROFILM	1	2	1	1	4
LISTS	0	4	0	3	4	DEFINITION	1	3	0	2	4
COMPACTING	1	2	1	2	4	OVERLAY	1	2	1	1	4
CDC	0	3	1	3	4	MONITORING	1	2	1	1	4
OUTLINED	0	4	0	3	4	TAPE	0	2	2	2	4
CREATION	1	1	2	3	4	ACCESS	1	1	2	3	4
LINC	0	3	1	1	4	FILING	0	2	2	1	4
AMESPLOT	1	3	0	1	4	QUICKSORT	1	2	1	1	4
GENERALIZA	0	4	0	1	4	OPTIMUM	1	2	1	1	4
DESIGNING	1	1	0	2	3	CAPABILITY	0	3	0	3	3
RELIABLE	1	2	0	1	3	TRANSMISSI	1	1	1	1	3
ERROR	0	0	3	2	3	BRAD	1	1	1	2	3
BROOKHAVEN	1	1	1	2	3	DISPLAYS	0	2	1	2	3
EMPLOYED	0	3	0	3	3	GENERATING	0	3	0	2	3
BINARY	0	2	1	2	3	ROTATING	0	1	2	2	3
SWEPT	0	1	2	1	3	DEVICE	1	1	1	3	3
DIFFICULT	0	3	0	3	3	COORDINATE	0	3	0	2	3
MAP	0	2	1	2	3	INEXPENSIV	1	1	1	3	3
BLIND	0	1	2	1	3	CONSTRUCTI	0	3	0	3	3
TACTILE	0	0	3	1	3	EMBEDDED	0	3	0	2	3
ONLINE	1	2	0	1	3	INCORPORAT	0	3	0	3	3
SEGMENTATI	0	1	2	3	3	DERIVED	0	2	1	3	3
MODIFICATI	1	2	0	2	3	LINEAR	0	1	2	1	3
CONTAINING	0	3	0	2	3	EFFICIENTL	0	3	0	2	3
ILLUSTRATI	0	2	1	3	3	AMBIGUITY	1	1	1	2	3
PROCESSORS	0	3	0	2	3	AUTOMATICA	0	2	1	3	3
ESTABLISHE	0	3	0	3	3	DIAGNOSTIC	0	2	1	3	3
UTILIZATIO	0	2	1	2	3	LINKAGE	0	1	2	2	3
COLLINS	1	2	0	1	3	RESULTANT	0	1	2	1	3
CONSIDERAT	0	3	0	3	3	SAMPLE	0	3	0	3	3
NUMERICAL	0	3	0	3	3	SUBROUTINE	0	2	1	2	3
FORTRAN	1	1	1	2	3	INTERFACES	0	2	1	2	3
LANGUAGES	0	2	1	2	3	CONCAVE	0	0	3	1	3
TELEVISION	0	1	2	2	3	RESULTING	0	2	1	2	3
STORED	0	2	1	3	3	REENTRANT	1	2	0	1	3
INTERRUPT	0	2	1	1	3	COMBINATIO	0	2	1	3	3
PLEX	0	1	2	1	3	STANDARDIZ	1	1	1	1	3
6000	0	2	1	2	3	BATCH	1	1	1	1	3
HUMAN	0	3	0	1	3	FACILITY	0	3	0	1	3
INSTRUMENT	1	0	2	2	3	CALCULATIO	1	1	1	1	3
HIERARCHY	0	2	1	1	3	LAP6	0	2	1	1	3
SCROLL	0	2	1	1	3	UNIFORM	0	3	0	2	3
SELF	0	2	1	1	3	SUBPLOTS	0	3	0	1	3
BOUND	0	3	0	1	3	RATIO	0	3	0	2	3
EQUATION	0	3	0	2	3	BRANCHING	0	3	0	2	3

REPRODUCIB	0	1	1	1	2	DEFINE	0	2	0	2	2
NOTION	0	2	0	1	2	SYNCHRONIZ	0	2	0	2	2
SEMANTICS	0	1	1	2	2	DETERMINIS	0	0	2	1	2
DEMONSTRAT	0	2	0	2	2	MULTICONSO	0	1	1	1	2
4000	0	2	0	2	2	DRUM	0	1	1	2	2
CONTINUOUS	0	2	0	2	2	COMPUTE	1	1	0	2	2
POINTING	0	2	0	1	2	SCREEN	0	1	1	2	2
BRAILLER	0	2	0	1	2	AID	0	1	1	2	2
LOCATING	0	2	0	2	2	STRINGS	0	2	0	2	2
ACCEPTS	0	2	0	1	2	PRODUCES	0	2	0	1	2
7094	0	2	0	2	2	PRESENTLY	0	2	0	2	2
UNSATISFAC	0	2	0	2	2	TYPING	0	2	0	1	2
PRIME	0	2	0	2	2	APPLICABLE	0	2	0	2	2
ILLUSTRATE	0	2	0	2	2	JUMPS	1	1	0	1	2
CASES	0	2	0	2	2	EASIER	0	1	1	1	2
ESTIMATE	0	1	1	1	2	SUM	0	2	0	2	2
PROPORTION	0	1	1	2	2	MATRICES	0	1	1	2	2
STEPWISE	1	1	0	1	2	REGRESSION	0	1	1	1	2
ALLOWS	0	2	0	2	2	ALTERNATIV	0	2	0	2	2
PERFORMED	0	2	0	2	2	PERFORM	0	2	0	2	2
HYPOTHETIC	0	2	0	2	2	CONSOLES	0	2	0	1	2
BOUNDS	0	2	0	2	2	OVERALL	0	2	0	2	2
ANSWERS	0	2	0	2	2	NONSTANDAR	1	1	0	1	2
GRAPHICALL	0	2	0	2	2	INPUTTING	0	2	0	1	2
CHINESE	0	1	1	1	2	TABLET	0	1	1	2	2
RELATIVE	0	1	1	2	2	COMPLETENE	0	2	0	1	2
CHECKING	0	2	0	1	2	DEFINED	0	2	0	2	2
PRODUCING	0	2	0	2	2	RICE	0	2	0	1	2
CONSECUTIV	0	2	0	2	2	LABELED	0	2	0	1	2
CONTAIN	0	2	0	2	2	CODEWORDS	0	1	1	1	2
BLOCKS	0	2	0	1	2	SYSTEMS	0	2	0	2	2
PROGRAMMED	0	2	0	2	2	ADDRESSING	0	1	1	1	2
FLEXIBILIT	0	2	0	2	2	EXECUTION	0	2	0	2	2
MULTI	0	2	0	2	2	EQUATIONS	0	1	1	1	2
SEQUENCE	0	1	1	2	2	DETAILED	0	2	0	2	2
FUNCTIONS	0	2	0	2	2	PROVISION	0	2	0	2	2
HALFTONE	0	1	1	1	2	HIDDEN	1	0	1	1	2
EDGES	0	0	2	1	2	SEQUENTIAL	0	1	1	2	2
STEREOGRAP	0	0	2	1	2	REFRESH	0	0	2	1	2
VIDEO	0	0	2	1	2	VIEWED	0	0	2	1	2
CREATING	0	1	1	2	2	INTERACTIV	1	0	1	2	2
RECOVERY	1	1	0	1	2	REFERENCE	0	2	0	2	2
NODES	0	2	0	1	2	REACHABLE	0	2	0	1	2
TRACING	0	1	1	2	2	KEYPUNCHED	0	2	0	1	2
INVOLVES	0	2	0	2	2	CODING	1	1	0	2	2
CONCURRENT	0	2	0	1	2	HANDLING	0	1	1	2	2
CHAINED	0	2	0	1	2	MACROS	0	2	0	2	2
IMPLEMENT	0	2	0	2	2	LOADER	1	0	1	1	2
GE	0	1	1	1	2	635	0	1	1	1	2
UNIVAC	0	1	1	1	2	1108	0	1	1	1	2
OPERATE	0	2	0	2	2	OCCUPATION	0	1	1	1	2
PARTITION	0	1	1	2	2	REMOTE	1	1	0	1	2
G21	0	2	0	1	2	EXISTING	0	2	0	1	2
RESIDENT	0	2	0	1	2	COMMANDS	0	2	0	1	2
INTERFACE	0	1	1	1	2	INTERACT	0	2	0	2	2

MTS	0	2	0	1	2	IDENTIFIED	0	2	0	1	2
RECORDED	0	2	0	1	2	SUPERVISOR	0	2	0	1	2
GUIDELINES	0	2	0	2	2	EVALUATION	0	0	2	2	2
ADVANCED	1	1	0	2	2	CRYPTOGRAP	1	1	0	1	2
CONFIDENTI	0	1	1	1	2	CYPTOGRAPH	0	1	1	1	2
SPECIALIZE	0	2	0	2	2	COMPILERS	0	1	1	1	2
SAVING	0	1	1	2	2	COMBINED	0	2	0	2	2
EQUIVALENC	0	1	1	1	2	MODIFY	0	2	0	2	2
PHRASE	0	2	0	1	2	GRAMMAR	0	1	1	1	2
FORMAL	0	2	0	2	2	SYNTACTIC	0	2	0	1	2
INDEXED	0	1	1	1	2	NUCLEUS	1	1	0	1	2
DIVERSE	0	2	0	2	2	HANDLED	0	2	0	2	2
UNIFORMLY	0	2	0	2	2	CONVERSATI	1	0	1	1	2
2048	1	1	0	1	2	PREPARATIO	0	2	0	1	2
ADDING	0	2	0	2	2	AIDS	0	2	0	2	2
TAPES	0	2	0	1	2	ORIENTED	0	0	2	1	2
DYNAMIC	0	1	1	1	2	MACHINES	0	1	1	1	2
INSIGHT	0	2	0	2	2	MULTICS	1	0	1	1	2
ARRAY	0	2	0	2	2	MEASURING	0	1	1	1	2
APPROPRIAT	0	2	0	2	2	PLOTTING	1	0	1	1	2
INSTALLATI	0	2	0	2	2	CONFIGURAT	0	2	0	2	2
SCALED	0	1	1	1	2	SCALING	0	2	0	1	2
TRANSFORMA	0	2	0	1	2	COMPARISON	0	1	1	1	2
HOARE	0	2	0	2	2	NONRECURSI	1	1	0	1	2
ORDERS	0	2	0	1	2	PARAMETERS	0	2	0	1	2
ROOTS	0	2	0	1	2	INTEGRAL	0	2	0	1	2
MOMENTS	0	2	0	1	2	SCAN	0	1	1	1	2
DESIGNER	0	1	0	1	1	ADOPT	0	1	0	1	1
ACCEPTING	0	1	0	1	1	REJECTING	0	1	0	1	1
CRITERIAOF	0	1	0	1	1	RECORDABIL	0	1	0	1	1
SPECIFIABI	0	1	0	1	1	ASYNCHRONO	0	1	0	1	1
IMPLY	0	1	0	1	1	DESIRES	0	1	0	1	1
INFLUENCES	0	1	0	1	1	AFFECTING	0	1	0	1	1
CONTENT	0	1	0	1	1	THECRITERI	0	1	0	1	1
ABSTRACT	0	1	0	1	1	CONCERN	0	1	0	1	1
CLOCK	0	1	0	1	1	INDEXES	0	1	0	1	1
REPEATABIL	0	0	1	1	1	PRIVACY	0	0	1	1	1
COMPUTINGR	0	0	1	1	1	TESTING	0	0	1	1	1
ELEPHONE	1	0	0	1	1	PROVEN	0	1	0	1	1
ACHIEVING	0	1	0	1	1	UNRELIABLE	0	1	0	1	1
FLOWCHART	0	1	0	1	1	INTERESTIN	0	1	0	1	1
TENS	0	1	0	1	1	THOUSANDS	0	1	0	1	1
CHARACTEER	0	1	0	1	1	FRAMES	0	1	0	1	1
INITIAL	0	1	0	1	1	INVESTMENT	0	1	0	1	1
PROGRAMMAT	0	1	0	1	1	DESIRED	0	1	0	1	1
HEADS	0	1	0	1	1	GENERATED	0	1	0	1	1
HORIZONTAL	0	1	0	1	1	SERVES	0	1	0	1	1
DRAWBACKS	0	1	0	1	1	INVERSE	0	1	0	1	1
CALCULATE	0	1	0	1	1	SHARED	0	1	0	1	1
REQUIRESA	0	1	0	1	1	DETAILS	0	1	0	1	1
33	0	1	0	1	1	MODIFYING	0	1	0	1	1
PRINT	0	1	0	1	1	RESILIENCY	0	1	0	1	1
PLATEN	0	1	0	1	1	EMBOSSER	0	0	1	1	1
SPECIFIC	0	1	0	1	1	ACOMPILER	0	1	0	1	1
COMPILER	0	1	0	1	1	SEARCHED	0	1	0	1	1

APPENDIX - D

Excerpts from the NCC thesaurus

2063 **11+ examination**
UF 2899 Eleven plus examination
BT 2861 Examinations

4024 **2c,l**
BT 3798 Part programs (machine tools)
NT 5363 2c,l applications

5363 **2c,l applications**
BT 4024 2c,l

3334 **4 bit codes**
BT 3004 Coded character sets

3770 **6 bit codes**
BT 3004 Coded character sets

3003 **7 bit codes**
BT 3004 Coded character sets

6792 **7 or 9 track magnetic tape transmission equipment**
BT 6065 Magnetic tape transmission equipment

4012 **7 track magnetic tape**
BT 2731 Magnetic tape
NT 6474 7 track magnetic tape encoders

6474 **7 track magnetic tape encoders**
BT 4012 7 track magnetic tape
1952 Magnetic tape encoders (stand alone)

7422 **80 column**
BT 7495 Attribute
4880 Punched card preparation bureaux

4861 **8 bit codes**
BT 3004 Coded character sets

7423 **96 column**
BT 7495 Attribute
4880 Punched card preparation bureaux

3772 **9 track magnetic tape**
BT 2731 Magnetic tape
NT 6475 9 track magnetic tape encoders

6475 **9 track magnetic tape encoders**
BT 3772 9 track magnetic tape
1952 Magnetic tape encoders (stand alone)

2374 **A/S regnecentralen**
UF 2375 Regnecentralen

2664 **Abbreviations (lists)**
BT 2900 Forms of publication

5860 **Abdominal diseases**
BT 6740 Abdominal systems
3923 Diseases
NT 5884 Appendicitis

6740 **Abdominal systems**
BT 2963 Anatomical systems
NT 5860 Abdominal diseases

4704 **Abelian groups**
BT 4705 Group theory

1125 **ABL applications**
Use 1216 Atlas basic language applications

933 **Abnormal loading**
BT 314 Loading
RT 1570 Dynamic loading
313 Static loading

1656 **Above surface handling equipment**
BT 1657 Carrying and lifting equipment
NT 1655 Cranes

7247 **ABS 1200 series**
BT 6277 Litton computers

6278 **ABS 1210**
BT 6277 Litton computers

6279 **ABS 1220/1221**
BT 6277 Litton computers

6280 **ABS 1231**
BT 6277 Litton computers

6849 **ABS 1231 applications**
BT 6848 Litton computer applications

6281 **ABS 1241**
BT 6277 Litton computers

6850 **ABS 1241 applications**
BT 6848 Litton computer applications

6282 **ABS 1251/1252**
BT 6277 Litton computers

6851 **ABS 1261 applications**
BT 6848 Litton computer applications

6283 **ABS 1281/1284**
BT 6277 Litton computers

369 **Absenteeism**
BT 2902 Offences (personnel)
NT 368 Absenteeism reports

368 **Absenteeism reports**
BT 369 Absenteeism
219 Management reports

3533 **Abstracting**
BT 3098 Information retrieval system operations
NT 3532 Automatic abstracting

3912 **Abstracting services**
BT 2900 Forms of publication
NT 2449 Chemical abstracts service

931 **Abutments**
BT 299 Supports
NT 932 Bridge abutments

6163 **Academic full-time courses**
BT 6161 Education (by length & place of study etc)

6164 **Academic part-time courses**
BT 6161 Education (by length place of study etc)

6165 **Academic short courses**
BT 6161 Education (by length place of study etc)

4819 **Accelerators (particle)**
Use 3047 Particle accelerators

2283 **Acceptance tests (computers)**
BT 4570 Computer acquisition
RT 2918 Computer projects
1096 Contracts (computer acquisition)
2522 Program validation
2387 Software verification
2230 Tenders (computer acquisition)

1 **Access methods**
Use 2 File organisation methods
3 Storage organisation methods

3522 **Accessioning**
BT 3149 Library materials processing

3765 **Accidents**
BT 3488 Hazards
NT 3764 Electrical accidents
4532 Road accidents

1273 **Account payments**
BT 107 Book-keeping

110 **Accountancy**
Use 108 Accounting

108 **Accounting**
UF 110 Accountancy
109 Accounts
684 Financial accounting
BT 836 Finance
NT 1871 Accounting (by application)
88 Accounting for exter appraisal
1559 Auditing
107 Book-keeping
754 Incomplete records accounting
1377 Invoicing
119 Management accour
1375 Remittance advice
RT 835 Rates collection

1871 **Accounting (by application)**
BT 108 Accounting
NT 6585 Freight accounting
4608 Inventory accounting
1447 Investment accounti
6949 Mailing subscription accounting
6584 Passenger accountin
190 Payroll administratic
1451 Purchase accounting
1145 Sales accounting
7329 Subscription accoun
6781 Vehicle costing
1146 Warehouse account

Accounting for external appraisal
BT 108 Accounting
NT 262 Assets
87 Balance sheets
397 Cash flow
1547 Expenditure analysis
385 Financial ratios
229 Financial statements
2675 Profit and loss

Accounting machines
Use 2645 Electronic accounting machines

Accounting machines (paper tape by-product)
BT 2645 Electronic accounting machines
6070 Punched paper tape preparation equipment

Accounts
Use 108 Accounting

Accounts payable
BT 111 Book-keeping records

Accounts reconciliation
BT 107 Book-keeping
NT 1140 Bank statement reconciliation

Ace computer
BT 1953 Computer history

Acid-base equilibrium
BT 131 Chemical equilibrium

ACM
Use 3937 Association for computing machinery

Acoustic couplers
UF 5315 Data couplers
5316 Telephone couplers
BT 3356 Modems
4157 Telephone sets

Acoustic devices
BT 4536 Mechanically operated devices
NT 4534 Acoustic generators

Acoustic generators
BT 4535 Acoustic devices
NT 4533 Buzzers

Acoustic measurements
BT 2670 Measurement
NT 2109 Acoustic wave analysers
RT 1084 Acoustics

Acoustic wave analysers
UF 2108 Sound analysers
BT 2110 Acoustic measurements
5435 Wave analysers
RT 2106 Voice recognition systems

Acoustics
BT 533 Physics
RT 2110 Acoustic measurements
5509 Solid state physics

Acquisition (library materials)
UF 3523 Ordering (library materials)
BT 3149 Library materials processing

4083 Activity sampling
BT 245 Work measurement

2674 Actuarial mathematics
BT 1513 Mathematics
NT 1530 Life tables
RT 3145 Insurance companies
417 Numerical analysis
390 Probability theory

5164 Actuators
BT 4834 Guidance components
4048 Servocomponents
5269 Valve components
NT 5286 Electric actuators
5347 Moving coil actuators

2755 Ad/four hybrid computer

9 Adams predictor corrector method
BT 10 Predictor corrector methods

802 Adaptive control systems
UF 2906 Self adaptive control systems
2907 Self adjusting control systems
2908 Self optimising control systems
BT 3008 Control system types
NT 3347 Impulsive response adaptive control systems
3348 Optimal control systems
801 Optimisation (control systems)
RT 4646 Adaptive logic
415 Automata
416 Automata theory
2581 Bionics
2580 Cybernetics
1463 Identification and modelling (control systems)
3403 Learning machines
4243 Self organising systems

4646 Adaptive logic
RT 802 Adaptive control systems

3305 Adders (digital computers)
BT 4289 Arithmetic units (by arithmetic process)
NT 4217 Binary adders
3655 Decimal adders
3654 Parallel adders
3653 Parallel decimal adders
3304 Serial adders

2187 Adding machines
BT 7355 Office equipment and activities
NT 6478 Adding machines (magnetic tape by-product)
6393 Adding machines (paper tape by-product)

6478 Adding machines (magnetic tape by-product)
BT 2187 Adding machines
1952 Magnetic tape encoders (stand alone)

6393 Adding machines (paper tape by-product)
BT 2187 Adding machines
6070 Punched paper tape

15 Address labels
Use 16 Labels

252 Addressing
BT 7497 Function
253 Instruction formats
NT 1962 Addressing (by medium)
254 Two-address formats

1962 Addressing (by medium)
BT 252 Addressing
NT 1963 Mass storage addressing

6656 Adhesives and gelatine industry
BT 6654 Miscellaneous chemical industries

6200 Adler computers
NT 6201 TA 100 series
7225 TA 10 series

6202 ADM business systems computers
NT 7264 P series
6203 Ricoh series

5083 Admissions (hospitals)
BT 3077 Hospitals

6158 Advanced courses
UF 6160 Specialist courses
BT 6155 Education (by depth of treatment)

3811 Advanced gas cooled reactors
UF 3812 AGR reactors
BT 3813 Carbon dioxide cooled graphite moderated reactors
3814 Gas cooled reactors
3815 Graphite moderated reactors

585 Advertising
BT 187 Marketing
NT 586 Advertising effectiveness
584 Media planning
RT 4401 Television advertising

586 Advertising effectiveness
UF 582 Coupon response
583 Media analysis
BT 585 Advertising
RT 584 Media planning
1011 Reader enquiry cards

5769 AED language
BT 3927 Algol extensions
5321 Computer aided design languages

2832 AEG-telefunken 60-50 computers
BT 3328 AEG-telefunken computers
2033 Process control computers

3327 AEG-telefunken
NT 3328 AEG-telefunken computers
3186 Telefunken computer gmbh

3328 AEG-telefunken computers
BT 3327 AEG-telefunken
NT 2832 AEG-telefunken 60-50 computers
RT 3187 Telefunken computers

2562 Aerial photography

Adaptive control systems
see 4137 Engineering

Adders (digital computers)
see 1661 Computers

Adding machines
see 7355 Office equipment and activities

Adding machines (magnetic tape by-product)
see 858 Data preparation
or 7355 Office equipment and activities
or 7386 Special purpose hardware

Adding machines (paper tape by-product)
see 858 Data preparation
or 7355 Office equipment and activities
or 7386 Special purpose hardware

Addo computers
This is a top term

Addo system 15
see 7224 Addo computers

Addressing
see 7494 Ancillary supplies/materials/services
or 253 Instruction formats

Addressing (by medium)
see 7494 Ancillary supplies/materials/services
or 253 Instruction formats

Adhesives and gelatine industry
see 3771 Industries

Adler computers
This is a top term

ADM business systems computers
This is a top term

Admissions (hospitals)
see 497 Medicine

Advanced courses
see 1706 Education and training

Advanced gas cooled reactors
see 4137 Engineering

Advertising
see 1647 Management

Advertising effectiveness
see 1647 Management

AED language
see 2671 Computer aided design
or 7568 Languages

AEG-telefunken 60-50 computers
see 3327 AEG-telefunken
or 1661 Computers
or 1706 Education and training
or 3771 Industries
or 2344 System architecture

AEG-telefunken
This is a top term

AEG-telefunken computers
see 3327 AEG-telefunken

5445 Aerial theory
see 4137 Engineering

5608 Aerial tracking
see 4137 Engineering

5340 Aerials
see 4137 Engineering

7284 Aero engine manufacturing and repairing industry
see 3771 Industries

1860 Aerodynamic characteristics
see 4137 Engineering
or 533 Physics

1859 Aerodynamic loading
see 4137 Engineering
or 533 Physics

5035 Aeronautical navigational charts
see 1666 Earth sciences
or 4580 Navigation

2827 Aerospace components
see 4137 Engineering

2180 Aerospace computers
see 1661 Computers
or 2179 Special purpose digital computers

1886 Aerospace engineering
see 4137 Engineering

2840 Aerospace industry
see 3771 Industries

5154 Aerospace structures
see 4137 Engineering

33 Aerospace transport
see 40 Transport

6530 Affix grammars
see 7568 Languages

2969 AFIPS
see 2925 Computer societies

2818 AFIPS conferences
see 2925 Computer societies
or 1706 Education and training

3136 Africa
see 1666 Earth sciences

6016 Age (sociology)
see 3727 Behavioral sciences

6595 Agricultural contracting industry
see 3771 Industries

2857 Agricultural machinery
see 2574 Agriculture

6116 Agricultural machinery dealers
see 2574 Agriculture

6675 Agricultural machinery manufacturing industry
see 2574 Agriculture
or 3771 Industries

2574 Agriculture
This is a top term

6593 Agriculture and horticulture industries
see 3771 Industries

1406 Air
see 1666 Earth sciences

3618 Air conditioning
see 4137 Engineering
or 7397 Environmental services

6103 Air conditioning equipment
see 4137 Engineering
or 7397 Environmental services
or 4473 Mining

24 Air core solenoids
see 4137 Engineering

1649 Air cored coils
see 4137 Engineering

26 Air cores
see 4137 Engineering

7220 Air cushion vehicles (hovercraft) industry
see 3771 Industries

4090 Air defence systems
see 4137 Engineering

3315 Air forces
see 4137 Engineering

4337 Air freight
see 4137 Engineering

5503 Air freighters
see 4137 Engineering

3248 Air pollution
see 2408 Ecology

39 Air routes
see 501 Operational research
or 40 Transport

2131 Air traffic control
see 4137 Engineering
or 993 Traffic control

7465 Air underground transit system
see 4137 Engineering

1681 Aircraft
see 4137 Engineering

7283 Aircraft & airframe manufacturing and repairing industry
see 3771 Industries

27 Aircraft (by name)
see 4137 Engineering

1680 Aircraft (by purpose)
see 4137 Engineering

5628 Aircraft (by take off)
see 4137 Engineering

5156 Aircraft (by wing characteristics)
see 4137 Engineering

1682 Aircraft engineering
see 4137 Engineering

2356 Aircraft industry
see 3771 Industries

5034 Aircraft navigation
see 4580 Navigation

5153 Airframes
see 4137 Engineering

7224 Addo computers
7223 . Addo system 15

6200 Adler computers
6201 . TA 100 series
7225 . TA 10 series
7295 . . B1728

6202 ADM business systems computers
7264 . P series
6203 . Ricoh series

3327 AEG-telefunken
3328 . AEG-telefunken computers
2832 . . AEG-telefunken 60-50 computers
3186 . Telefunken computer gmbh
3187 . . Telefunken computers
3188 . . . . Telefunken computer applications
1122 . . . . TR 4 applications
6035 . . . . TR 86 applications
5867 . . . Telefunken TR 440

2574 Agriculture
2857 . Agricultural machinery
6116 . . Agricultural machinery dealers
6675 . . Agricultural machinery manufacturing industry
5672 . Agronomy
5670 . . Agronomy (by products)
5667 . . . Farm crops
7353 . . . . Fruits
5758 . . . . . Bananas
5666 . . . . Grain crops
5665 . . . . . Wheat plants
3253 . Farm management
4520 . Fisheries
4519 . . Fishing gear
5299 . Forestry

6204 Allied business systems computers
6969 . Allied business systems computer applications
6205 . GRI-909 computer
6206 . GRI-99 computer
6207 . Multibus computer system

3020 Analog-digital conversion
2359 . . Digitisers
4845 . Shaft position converters

7494 Ancillary supplies/materials/services
7495 . Attribute
7422 . . 80 column
7423 . . 96 column
7654 . . Ancillary/auxillary
7500 . . Drop resistant
7501 . . Edge punched
7503 . . Envelopes
7499 . . . Fire resistant
16 . . Labels
7502 . . Ledger
6531 . . Optical character readers
5030 . . Optical mark readers
7504 . . Ribbons
7497 . Function
252 . . Addressing
1962 . . . Addressing (by medium)
1963 . . . . Mass storage addressing
254 . . . Two-address formats
251 . . . . Two-address compilers
7520 . . Bursting
7519 . . Cleaning
5317 . . Collators
7521 . . Decollating
7664 . . Dispensing
7663 . . Duplicating
7528 . . Environmental
7668 . . Filing
7669 . . Flooring
7522 . . Guillotining
7526 . . Labelling
7665 . . . Mailing
7524 . . Microfilming
7661 . . Power supply
1898 . . Printing
7068 . . . Forms printing
7407 . . . . Computer stationery printing
7406 . . . . Optical character recognitions-forms design/printing
7405 . . . . Optical mark recognition forms design/printing
1897 . . . Printing operations
1896 . . . . Composition
1448 . . . . . Typesetting
2763 . . . . . Photocomposition
3744 . . . . Estimation (printing)
4903 . . . . Proof correction
3199 . . . Printing processes
3198 . . . . Planographic printing
3197 . . . . . Illustrations (printing)
3196 . . . . . Half tones
3671 . . . . . Lithography
3331 . . . . . Offset lithography
2804 . . . . . Diography
7662 . . Reading
7523 . . Shredding
7525 . . Site preparation
7527 . . Splicing
7667 . . Splicing
7592 . . Storage
336 . . Testing
7565 . . . Avionic equipment testing
1759 . . . Chemical analysis
3685 . . . Chemical analysis processes
2856 . . . . . Chromatographic analysis
5429 . . . . . Gas chromatographic analysis
6590 . . . . . Elemental analysis (organic)
6589 . . . . . Biochemical analysis
5415 . . . . Optical chemical analysis
5413 . . . . . Optical emission chemical analysis
5412 . . . . . Fluorimetry
5411 . . . . . Particle fluorimetry
5410 . . . . . X ray fluorimetry
5575 . . . . . Spectroscopy
5574 . . . . . Spectroscopes
1760 . . . . . Particle physics chemical analysis
1761 . . . . . Radiochemical analysis
1762 . . . . . Neutron activation analysis
5407 . . . . . X ray diffraction
5410 . . . . . X ray fluorimetry
5541 . . . . . Volumetric analysis
5540 . . . . . Titration (electrical)
5539 . . . . . Potentiometric titration
6777 . . . . Water analysis
5825 . . . Flight tests
329 . . . Mechanical testing
331 . . . . Destructive testing
332 . . . . . Compression testing
266 . . . Program testing
267 . . . . Debugging

(continued)

7494 Ancillary supplies/materials/services *(continued)*
265 . Test data
264 . . Test data generators
1194 . . . Trace programs (continuous)
333 . . . Test equipment
2305 . . . Electronic test equipment
4174 . . . Rigs
4173 . . . . Tunnels (rigs)
4172 . . . . . Wind tunnels
335 . . Test cubes
3697 . . . Testing techniques
2795 . . . Automatic testing
3501 . . . . . Automatic test languages
2304 . . . . Programmed test equipment
5194 . . . . . Storage testers
6084 . . . Experimental stress analysis
6083 . . . . Photoelastic stress analysis
7666 . . Winding
7498 . Main class
7530 . . Equipment
593 . . . Materials
594 . . . Compounds
595 . . . . Minerals and ores
6741 . . . . Organic compounds
3610 . . . . . Alcohols
5856 . . . Dangerous materials
5855 . . . . Mine gases
5218 . . . . Radioisotopes
6497 . . . . . Xenon
6496 . . . . . Xenon-133
4071 . . . Elements
4070 . . . . Metallurgy
4948 . . . . Metals
5570 . . . . . Metal ions
5883 . . . Ferroelectric materials
5882 . . . . Potassium nitrate
4403 . . Materials (by manufactured form)
455 . . . . Composite materials
785 . . . . . Reinforced materials
780 . . . . . Reinforced concrete
895 . . . . . Reinforced concrete structures
813 . . . . . Reinforced concrete beams
6050 . . . . . Reinforced concrete box culverts
1803 . . . . . Reinforced concrete chimneys
789 . . . . . Reinforced concrete columns
1265 . . . . . Reinforced concrete continuous beams
897 . . . . . Reinforced concrete foundations
898 . . . . . Rectangular reinforced concrete foundations
919 . . . . . Reinforced concrete frames
4717 . . . . . Reinforced concrete horizontal cantilever wing
1331 . . . . . Reinforced concrete plane frames
6391 . . . . . Reinforced concrete reservoirs
1518 . . . . . Reinforced concrete sections
814 . . . . . Reinforced concrete slabs
781 . . . . . Reinforcing materials
784 . . . . . Reinforcing steels
2916 . . . Materials (by physical property)
1389 . . . . Anisotropic materials
6009 . . . Materials (by purpose)
6008 . . . . Drugs
4502 . . . . . Antibiotics
5019 . . . . . Kanamycin
6006 . . . . . CNS depressants
6005 . . . . . Anaesthesics
6004 . . . . . Halothane
6752 . . . . . Haematological drugs
5249 . . . . . Anticoagulants
7529 . . Service
7531 . . Supply
7496 . Media
7517 . . Binder
7513 . . Cabinet
7507 . . Card
7655 . . Cardboard
7512 . . Cartridge
7511 . . Cassette
3869 . . Continuous stationery
7518 . . Document
7505 . . Magnetic disc
7506 . . Magnetic disc(floppy)
2731 . . Magnetic tape
7657 . . Magnetic tape
7658 . . Magnetic ticket
2594 . . Microfilm
4926 . . . Microfilm equipment suppliers
3211 . . . Microfilm files
2797 . . . Pcmi
7656 . . Mylar tape
7516 . . Pack
7509 . . Plastic card
7508 . . Plastic tape
2766 . . Punched paper tape
4925 . . . Punched paper tape storage units
4924 . . . . Punched paper tape storage unit suppliers
7510 . . Reel/spool
7514 . . Safe
7515 . . Tray

5978 **Anti-skid devices**
5970 . Brakes
5967 . . Brakes (by mode of operation)
5964 . . . Pneumatic brakes

1715 **Applications**
370 . Business applications
371 . . Business time sharing
7569 . . Mode of operation
2010 . . . Batch processing
1972 . . . . Remote batch processing
3204 . . . . . Remote batch terminals
213 . . . Interactive computing
2026 . . . . Interactive computing applications
2362 . . . . . Interactive computer graphics
3553 . . . . . Interactive information retrieval
1999 . . . . . Interactive least squares data fitting
2000 . . . . . Interactive mathematics
1578 . . . . Interactive languages
2488 . . . . . APL
7323 . . . . . APL applications
2683 . . . . . APL/360 applications
2654 . . . . . Basic (language)
1227 . . . . . Basic applications
1553 . . . . . Basic compilers
1552 . . . . . Basic programs
3601 . . . . . Joss
4404 . . . . . Telcomp language
4403 . . . . . Telcomp programs
1997 . . . Real time systems
1973 . . . . Data transmission
5239 . . . . . Data transmission codes
5238 . . . . . Pseudoternary codes
5314 . . . . . Data transmission equipment
4702 . . . . . Data concentrators
6064 . . . . . Facsimile transmission equipment
6065 . . . . . Magnetic tape transmission equipment
6792 . . . . . 7 or 9 track magnetic tape transmission equipm
6791 . . . . . Magnetic tape cassette transmission equipment
3356 . . . . . Modems
4191 . . . . . Acoustic couplers
6793 . . . . . High frequency data links
6794 . . . . . Infrared communication links
4316 . . . . . Matrix printers
6797 . . . . . Microwave data links
6798 . . . . . Radio wave data links
6066 . . . . . Punched card transmission equipment
6067 . . . . . Punched paper tape transmission equipment
2215 . . . . . Terminals
2216 . . . . . Display devices
5923 . . . . . Digital displays
2965 . . . . . Display devices (by component)
2162 . . . . . Cathode ray tube displays
6486 . . . . . Cathodochromic displays
4528 . . . . . Storage tube displays
2739 . . . . . Holographic displays
6060 . . . . . Graphical displays
5343 . . . . . Light pens
6194 . . . . . Rear projection screens
2598 . . . . . Touch displays

1715 Applications *(continued)*
7105 . . . . . Video terminals
203 . . . . . Visual display units
3365 . . . . . Stereoscopic displays
2407 . . . . . Data transmission networks
3229 . . . . . Data transmission systems
3500 . . . . . Data links
6793 . . . . . High frequency data links
6794 . . . . . Infrared communication links
6797 . . . . . Microwave data links
6798 . . . . . Radio wave data links
3228 . . . . . Telegraphy
5099 . . . . . Telegraph apparatus
5096 . . . . . Telegraph switches
5092 . . . . . Crossbar switches
3227 . . . . . Telegraph transmission systems
3226 . . . . . Telegraph lines
2087 . . . . Highway (data transmission)
5711 . . . . . Serial data transmission
3862 . . . . . Start/stop transmission
3863 . . . . . Synchronous transmission
1583 . . . . . Teleprocessing
2224 . . . . Message switching systems
2235 . . . . Reservation systems
2234 . . . . . Hotel reservations
2054 . . . . . Seat reservations
2055 . . . . . Ticketing
7435 . Industry applications
7569 . . Mode of operation
2010 . . . Batch processing
1972 . . . . Remote batch processing
3204 . . . . . Remote batch terminals
213 . . . Interactive computing
2026 . . . . Interactive computing applications
2362 . . . . . Interactive computer graphics
3553 . . . . . Interactive information retrieval
1999 . . . . . Interactive least squares data fitting
2000 . . . . . Interactive mathematics
1578 . . . . Interactive languages
2488 . . . . APL
7323 . . . . . APL applications
2683 . . . . . APL/360 applications
2654 . . . . . Basic (language)
1227 . . . . . Basic applications
1553 . . . . . Basic compilers
1552 . . . . . Basic programs
3601 . . . . . Joss
4404 . . . . . Telcomp language
4403 . . . . . Telcomp programs
1997 . . . Real time systems
1973 . . . . Data transmission
5239 . . . . . Data transmission codes
5238 . . . . . Pseudoternary codes
5314 . . . . . Data transmission equipment
4702 . . . . . Data concentrators
6064 . . . . . Facsimile transmission equipment
6065 . . . . . Magnetic tape transmission equipment
6792 . . . . . 7 or 9 track magnetic tape transmission equipment
6791 . . . . . Magnetic tape cassette transmission equipment
3356 . . . . . Modems
4191 . . . . . Acoustic couplers
6793 . . . . . High frequency data links
6794 . . . . . Infrared communication links
4316 . . . . . Matrix printers
6797 . . . . . Microwave data links
6798 . . . . . Radio wave data links
4066 . . . . . Punched card transmission equipment
6067 . . . . . Punched paper tape transmission equipment
2215 . . . . . Terminals
2216 . . . . . Display devices
5923 . . . . . Digital displays
2965 . . . . . Display devices (by component)
2162 . . . . . Cathode ray tube displays
6486 . . . . . Cathodochromic displays
4528 . . . . . Storage tube displays
2739 . . . . . Holographic displays
6060 . . . . . Graphical displays
5343 . . . . . Light pens
6194 . . . . . Rear projection screens
2598 . . . . . Touch displays
203 . . . . . Visual display units
3365 . . . . . Stereoscopic displays
2407 . . . . . Data transmission networks
3229 . . . . . Data transmission systems
3500 . . . . . Data links
6793 . . . . . High frequency data links
6794 . . . . . Infrared communication links
6797 . . . . . Microwave data links
6798 . . . . . Radio wave data links
3228 . . . . . Telegraphy
5099 . . . . . Telegraph apparatus
5096 . . . . . Telegraph switches
5092 . . . . . Crossbar switches
3227 . . . . . Telegraph transmission systems
3226 . . . . . Telegraph lines
2087 . . . . Highway (data transmission)
5711 . . . . . Serial data transmission
3862 . . . . . Start/stop transmission
3863 . . . . . Synchronous transmission
1583 . . . . . Teleprocessing
2224 . . . . Message switching systems
2235 . . . . Reservation systems
2234 . . . . . Hotel reservations
2054 . . . . . Seat reservations
2055 . . . . . Ticketing
7045 . . Public utilities
1716 . Machine applications
6969 . . Allied business systems computer applications
2168 . . Analog computer applications
2933 . . Burroughs computer applications
7002 . . . B1700 applications
7575 . . . . B1714 applications
1293 . . . B200 applications
6547 . . . . B263 applications
6691 . . . . B270 applications
6692 . . . . B273 applications
1294 . . . . B283 applications
7056 . . . B2700 applications
1566 . . . B300 series applications
6550 . . . . B363 applications
6548 . . . . B370 applications
6549 . . . . B373 applications
1134 . . . . B383 applications
1295 . . . . B385 applications
7368 . . . B3200 applications
7300 . . . B3700 series applications
2221 . . . B500 series applications
1405 . . . . B2500 applications
4988 . . . . B2506 applications
1221 . . . . B3500 applications
6693 . . . . B4500 applications
6545 . . . . B500 applications
2220 . . . . B502 applications
1567 . . . . B5500 applications
1568 . . . . B6500 applications
2940 . . . B700 series applications
4807 . . . . B4700 applications
2435 . . . . B6700 applications
7025 . . . Burroughs I series applications
2570 . . . Burroughs tc500 applications
3369 . . . Burroughs tc600 applications
5105 . . . E6000 applications
6551 . . . E3000 applications
5106 . . . E4000 applications
5107 . . . E6000 applications
6546 . . E8000 applications
4385 . . Business computers ltd computer applications
7251 . . . Molecular 18 applications
7620 . . . Molecular 6 applications
4384 . . Susie applications
1719 . . CDC computer applications
1736 . . . CDC 1604 applications
1297 . . . CDC 160a applications
2052 . . . CDC 1700 applications
6942 . . . CDC 1700 series applications
2052 . . . . CDC 1700 applications
6904 . . . . CDC 1704 applications
6905 . . . . CDC 1774 applications
1720 . . . CDC 3000 series applications
1298 . . . . CDC 3100 applications